InriaSoft Presentation

David Mergery presented InriaSoft, an Inria foundation that will manage most of the Inria consortiums. At the beginning, 5 consortiums should be involved: SOFA, MUMPS, Pharo, Coq, mmgtools. InriaSoft should be located in Rennes.

Discussion
Is it a good idea to separate the consortiums from their corresponding research team? SOFA should rather be located in Lille (Defrost) or in Strasbourg (Mimesis)?
→ It has been planned this way for financial reasons. Discussions to come about this with the other consortiums. Still open to discussion.

The Consortium Report

The Consortium presented its last 6 month activities and what is planned for the next 6 months. See the slides for more details.

What has been done

- Management
  - Consortium contract v1: now expecting the first memberships
- Community
  - Events: MMVR Los Angeles, GTAS Paris, IGG Strasbourg, ...
  - Contacts: academics & industrials
  - Do not hesitate to ask the Consortium for contacts!
- Website
  - Gallery & Marketplace: new items added. Provide us even more content!
  - Open documentation: community tasks to improve the documentation
- Framework
  - GitHub migration
  - v16.08 release (>250 downloads of binaries)
What’s next

- Management
  - Administrative Board: design the future offer of the Consortium
  - Consortium contract v2

- Community
  - Events: SOFA Days, FOSDEM Bruxelles, I-Novia Strasbourg, …
  - Open dev workflow
  - Let us know for common events / shared booth!

- Website
  - Marketplace growing: help us promote your work

- Framework
  - RESET coding sprint 2
  - GitHub issues/PR handling
  - Tutorial application: any first user starts by opening the Modeler to launch the tutorials. Need of a dedicated application for tutorials
Workgroup discussions

The rest of the STC was dedicated to discussions about past and next tasks around SOFA on these themes: programming, quality, documentation, contributions, strategy. The goal was to isolate 4 important tasks to be achieved (or at least well advanced) in 6 months, and then to build Workgroups of 5-6 persons around these tasks. Added to the Workgroups, we also defined some background tasks: important tasks which do not need a Workgroup.
All these tasks define the SOFA Roadmap for the next 6 months.

GitHub migration: TASK OVER

- GForge repository has been migrated to GitHub. One modification has been made to the architecture: SofaKernel.
- We discussed about splitting in two repositories (SOFA + SofaKernel) but this can be difficult to manage (branching, …) and does not bring much value.
- For now, we will remain with one repository.

Logger documentation: TASK OVER

- Doc page created and added to online documentation.
To be improved.

C++11: TASK OVER

- C++11 is now mandatory in SOFA.
- Compilers compatibility: the continuous integration for GCC 4.6 and VS2012 will be stopped (C++11 coverage issues). The new configuration will be: Clang 3.4, GCC 4.8, VS2013, VS2015.
- Moreover, we have to specify which C++11 features we use (some compilers do not support all features) in CMake, this will print warnings if not respected.

GUI: TASK OVER

- The first version is done. It is called SofaQtQuick and licenced under GPL. It is composed of a plugin (SofaQtQuickGUI) and a project (qtQuickSofa). It brings more langages to learn (Javascript) but runSofa is really broken, its main function is a mess and GUI Loading (which GUI we chose) is hard coded.
- Therefore, we encourage all devs to use it / test it. It may replace runSofa in a near future but it still needs to be polished.

→Ask Olivier Carre (carre@anatoscope.com) for repository access.
Dynamic topologies: CONTINUED AS BACKGROUND TASK

- Here is a summary of Etienne’s presentation.
  It first started with the work that has been done since the beginning of the project
  - MapTopology, C++11 replacement for BaseMeshTopology
  - New components using this interface
  - New cell traversal
  - Compatibility components
- Issues faced
  - Each component needs to be rewritten to benefit for combinatorial maps.
    It corresponds to a huge amount of work.
- Solutions
  - “bridge” component
- Remaining
  - a working example of the new interface (cutting plugin)
  - push the code in a branch of the SOFA public repository
- Questions from the audience
  - Efficiency? 3x to 3.5x faster with complex functions
  - Parallelism done how? thread pool, no boost dependency, C++11 and standard lib
  - Name consistency with SOFA? Work in progress.
  - CGoGN stable? Yes. Rewriting cutting plugin because going from CGoGN 1.0 to 2.0 (more stable)
  - Split between the different use cases (to NOT have to much things in Topology class)? Not done yet.

- A demonstration with real comparison and some tests are required to engage the missing manpower. Making this work more public was also a query of the participants.

- Conclusion: till then, this task is not anymore monitored by the Consortium. We advise more communication about the topic and propose a demo app to illustrate the benefits of CGoGN with SOFA. A GitHub could also be created to spread the word.
  Contact: etienne.schmitt@inria.fr

Python binding: CONTINUED AS WORKGROUP

- Comparison SWIG vs Cython: implement the same binding for a component with both approaches: which was the best?
  Results: Cython more efficient, SWIG more used/supported.
- Increase the granularity of the system by bypassing the SOFAPython mutex lock.
  Doable with SWIG?
Main problem: What do people want to do with Python in SOFA? Create scenes (we will go to simple api) or work in the simulation loop (more complex needs)?

Questions
- How deep do we want python in SOFA?
- Keep using XML + Python? Only Python? If we use Python for writing scenes (better than XML), no XML means no more tool to create scenes (Modeler).
  - Who is using the Modeler? Who is using XML?
  - We should ask, make people vote

first contact: Bruno Carrez
members: Matthieu Nesme, Damien Marchal, Etienne Schmitt, Bruno Marques, Christoph Paulus
Full details on GitHub: https://github.com/sofa-framework/sofa/issues/23

Mappings and Masks: CONTINUED AS WORKGROUP

Problem of Masks: it affects everything with the mechanicalstate, real issue: enable only when we want it
Participants agreed to extend this workgroup to all Mappings.
Discussions about two applyJ with dense and sparse versions (getJ).
Constraints: issues when the applyJT function is used

first contact: Christian Duriez (and Matthieu Nesme)
members: Jeremie Allard, Francois Faure, Hadrien Courtecuisse, Eulalie Coevoet, Igor Peterlik, Francois Jourdes
Full details on GitHub: https://github.com/sofa-framework/sofa/issues/26

Multi-threading: CONTINUED AS WORKGROUP

See enclosed slides of TruPhysics.
Work has been done around multi-threaded collision detection:
- concerning the pipeline of SOFA: not too many api changes
- multi threaded traversals of the scene graph
- separate simulation and visu loops

InSimo has patches that can solve some of the mentioned problems (will be merged in SOFA by RESET sprint). What they have to put in the core of SOFA is the fixes making possible to implement a threaded solution (depends on what you are doing in your animation loop), not to impose one.
Component threaded approach works well with scenes with a lot of objects

first contact: Fabian Aichele
members: Francois Faure, Christian Duriez, Thomas Lemaire, Igor Peterlik, Hadrien Courtecuisse, Hervé Delingette
Full details on GitHub: https://github.com/sofa-framework/sofa/issues/24
Validation: STARTED AS WORKGROUP

Validation would be nice to attract people (students, industrials).
→ compare with industrial solutions: simpy comparison

first contact: Igor Peterlik
members: Rémi Bessard, Hervé Delingette, Damien Marchal, Frédérick Roy, Erik Pernod, Jeremie Allard
Full details on GitHub: https://github.com/sofa-framework/sofa/issues/25

Parameter update: CONTINUED AS BACKGROUND TASK

- The idea is to get and use up-to-date parameters and matrices. The case of sensitivity analysis requires the derivatives regarding parameters (Igor). A deep level engine is required to enforce consistency between datas→ DataEngine (Matthieu).
- The update mechanism could be handled by only one component dealing with all updates would this be applied for the initData (dependencies at the init)
- Matthieu considered it as a small task that he could handle. The amount of work should not need a 6 month workgroup. This will therefore be continued as a background task. It also should be done in coordination with InSimo, since Francois Jourdes would like to give some input about this topic as well.

contact: Matthieu Nesme
member: François Jourdes, Hugo Talbot
Matthieu will update us on this topic. Maybe a GitHub issue could be created around this topic.

Cleaning & Quality: CONTINUED AS BACKGROUND TASK

- Objective: make SOFA cleaner.
  The idea is to remove what is unused or duplicated in SOFA, fix what is broken.
- Setup some tools/processes to improve quality in existing code and in new contributions.
  e.g. : existing static analysis can be found in tools/
- All the lines of core code should be parsed by a test → 100% code coverage
  This should be a background task for everyone.
- How to motivate people to do this? Team leaders have to motivate their SOFA devs.

first contact: Consortium (Guillaume)
A GitHub issue will be created soon.
Clean CMake: PAUSED

- This task will be continued when working on a smaller/cleaner SOFA.
- The main CMake has to be as simple as possible: we need a simple way to add extlibs/plugins in SOFA.
- The “make install” must be tested to see if there are lib finding/linking issues (findX.cmake)
- Add coding guidelines in cmake so that everything works well.

first contact: Frederick Roy

SOFA as a lib

- This was a discussion.
- Pushed by the community, developers are working on it.
- To present SOFA as a lib, we should create an API that is similar to other physics engines (like Bullet).

Future of SOFA

- This was a discussion.
- First, we want to make SOFA easier to use. We would like to move towards a lighter version of SOFA, thus making it easier to use and start with.
- About the repository architecture: splitting repo (Kernel stable) can be difficult to manage (branching, …) and does not bring much value yet.
# STC#2 Workgroups selection

<table>
<thead>
<tr>
<th>Workgroups</th>
<th>Members</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multithreading</td>
<td><strong>Fabian</strong>, Stéphane, J1, FJ, Thomas</td>
<td><a href="https://github.com/sofa-framework/sofa/issues/24">https://github.com/sofa-framework/sofa/issues/24</a></td>
</tr>
<tr>
<td>Masks/Mappings</td>
<td><strong>Christian</strong>, Matthieu, FF, FJ, J1, Olivier, Eulalie</td>
<td><a href="https://github.com/sofa-framework/sofa/issues/26">https://github.com/sofa-framework/sofa/issues/26</a></td>
</tr>
</tbody>
</table>

Find all workgroup discussions on GitHub: [https://github.com/sofa-framework/sofa/issues?utf8=%E2%9C%93&q=label%3Adiscussion%20label%3Aworkgroup](https://github.com/sofa-framework/sofa/issues?utf8=%E2%9C%93&q=label%3Adiscussion%20label%3Aworkgroup)