SPIS-GEO preliminary study

Feasibility study of SPIS adaptation to geostationary orbits and to its usage in industrial context

Benoît Thiébault[(1)], Julien Forest[(2)]
Contact : thiebault@artenum.com
Supervised by Denis Payan[(3)]
CNES Contract: DAJ/AR/PS N°08-9852
(1)Artenum, Toulouse
(2)Artenum, Paris
(3)CNES

Outline

- Objectives of the study
- SPINE community
- Users feedback
- SPIS-UI architecture analysis
- Software requirements
- Platform migration
- Community life and business models

Objectives of the study

- Objectives:
  - Define SPIS future evolutions in order to increase the use of SPIS in the industry:
    - Simplify usage for non-experts users
    - Extension to industry-specific issues (Geostationary orbit, modern and complex commercial platform, electric propulsion)
- Two steps:
  - Gather SPIS users feedback and future users wishes
    - Community analysis (forum, survey) and interviews
    - Industrial users (and non-users) interviews
  - Define new requirements based on:
    - Users feedback
    - Bibliographic studies
    - Existing codes overview and critical analysis
    - Technological watching and analysis
    - Prototyping and tests

Study output

- A consistent set of documents:
  - Deliverable #1 / Users feedback and application scenarios
  - Deliverable #2 / Numerical models identification report
  - Deliverable #3 / SPIS architecture analysis
  - Deliverable #4 / SPIS-GEO software requirements
  - Deliverable #5 / Platform migration analysis
  - Deliverable #6 / Client/Server or Service Oriented Architectures analysis
  - Deliverable #7 / Effort and budget estimates
  - Deliverable #8 / SPINE community management strategy and business models
  - Deliverable #9 / Final Report

Outline

- Objectives of the study
- SPINE community
- Users feedback
- SPIS-UI architecture analysis
- Software requirements
- Platform migration
- Community life and business models

SPINE Community

- http://www.spis.org
- About 219 registered persons and a large ten of active members
- SPINE meeting every 6 months
- An active forum
  - 90 threads (i.e. subjects)
  - 250 message in less than 42 months
  - Average 6 messages a month.
- Downloads
  - 3.1 (all versions): about 185
  - 3.6 (all versions): about 347
  - 3.7 (all versions): about 80 (no announcement done)

Outline

- Objectives of the study
- SPINE community
- Users feedback
- SPIS-UI architecture analysis
- Software requirements
- Platform migration
- Community life and business models

Users feedback (1/5)

- Usage
  - Majority of basic users
  - Mainly used for mission design and theoretical modeling

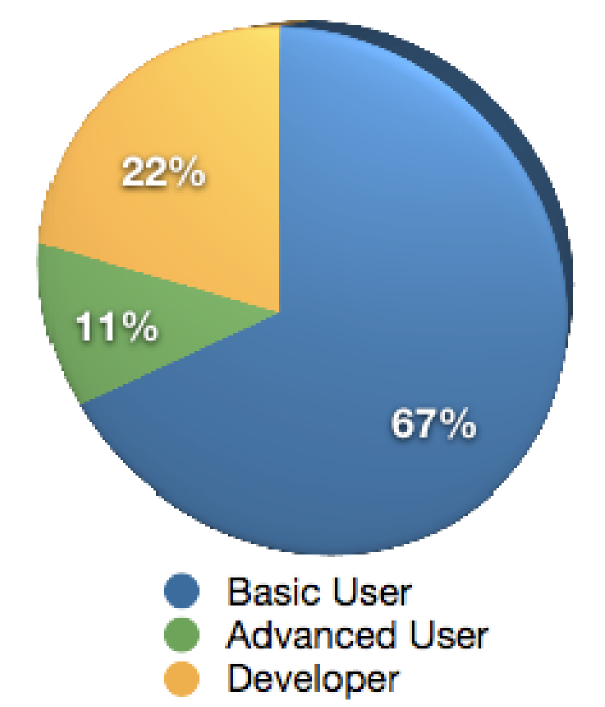
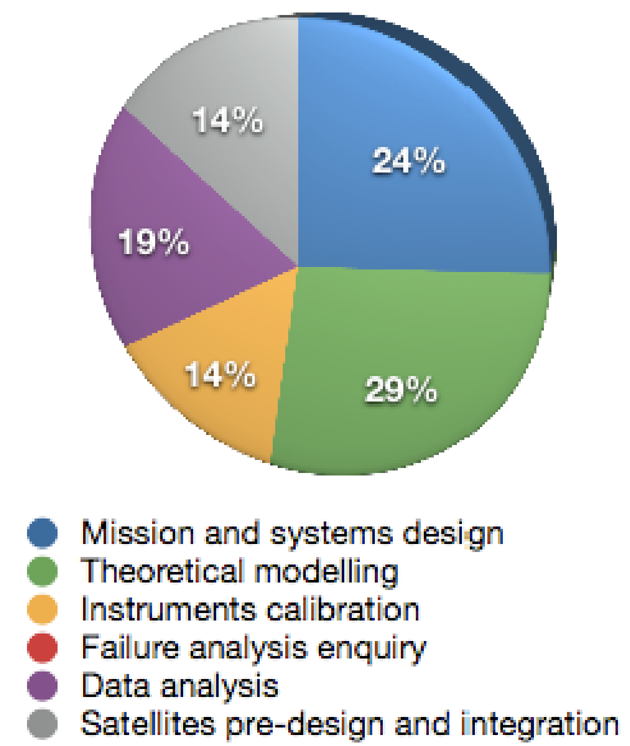


Figure 2: Kind of users



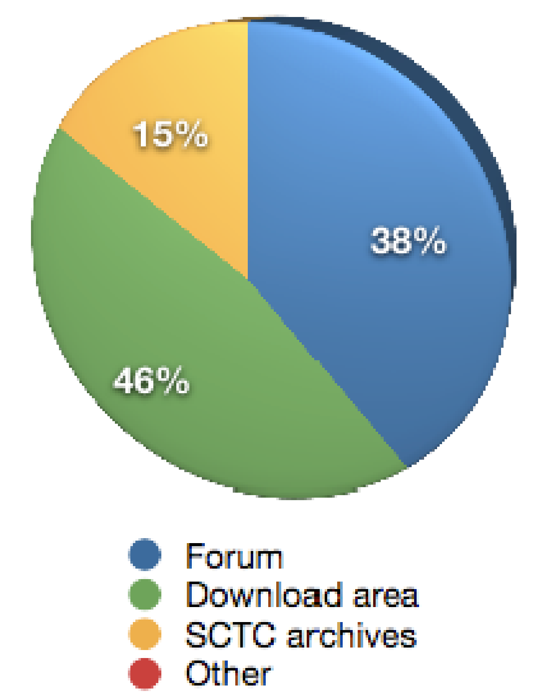


Figure 14: SPINE website usage

Users feedback (2/5)

- SPIS 3.6 rating and users wishes
  - Perceived stability and ease of use (especially for pre-processing) are the main demands for evolutions
  - Pre-defined simulations for the most relevant cases are very important
  - Need of scenarios
  - Interoperability with other tools is not a priority for the users

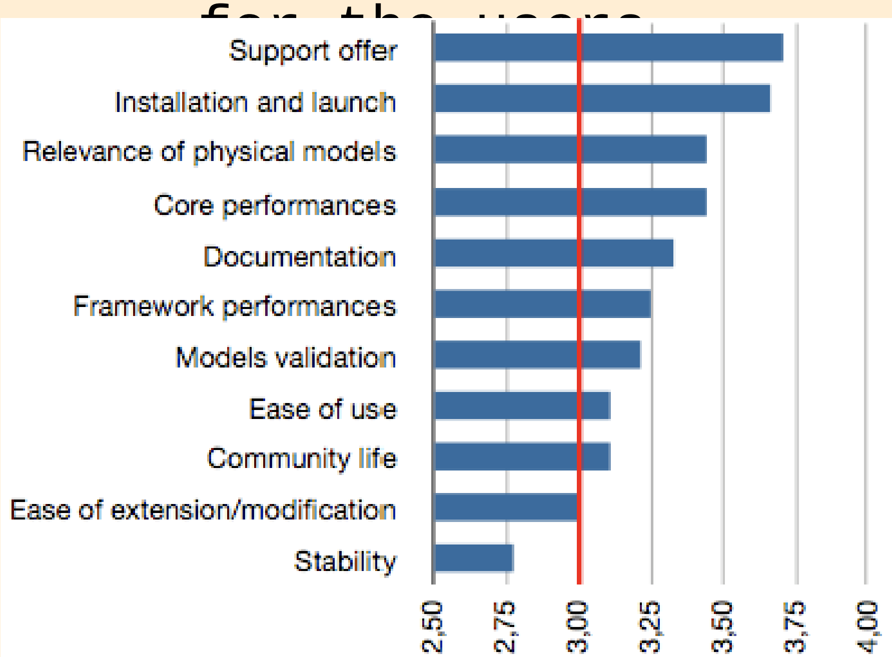


Figure 4: SPIS functionalities and services rating by users

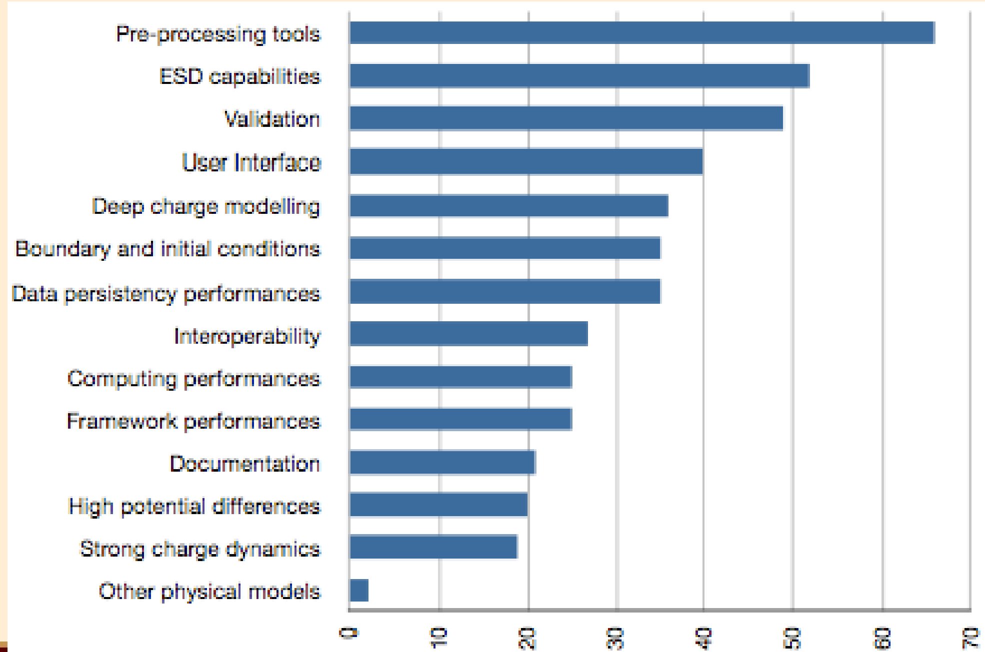


Figure 5: Highest rated user requirements (Condorcet method scale)

Caution: evaluation done on the basis of SPIS 3.6

Users feedback (3/5)

- Remote access and computation speed
  - SPIS is still mainly used locally on basic desktop computers.
  - However, increasing use of SPIS in both modes (local, client-server)
    - Increasing interest use of SPIS in Intranet client-server mode
    - Web-based version is not a priority but an interesting future
  - The computation speed ... ct (r... s than 1/2...

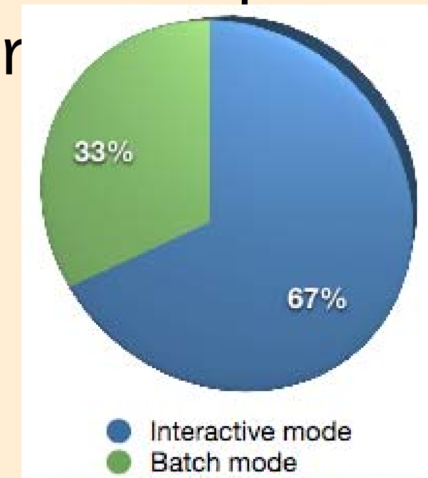


Figure 11: Batch / interactive mode

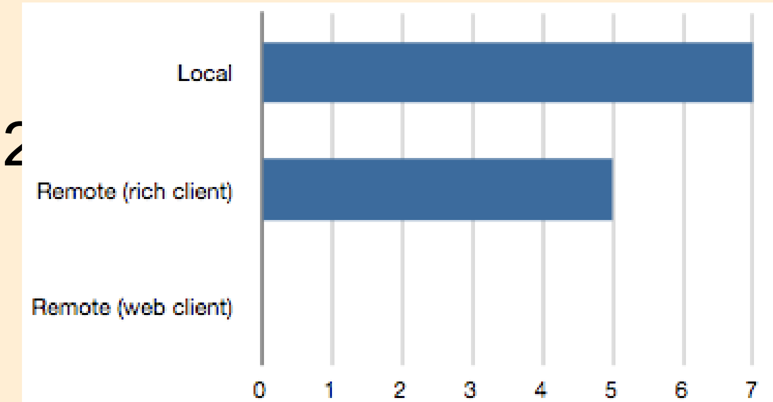
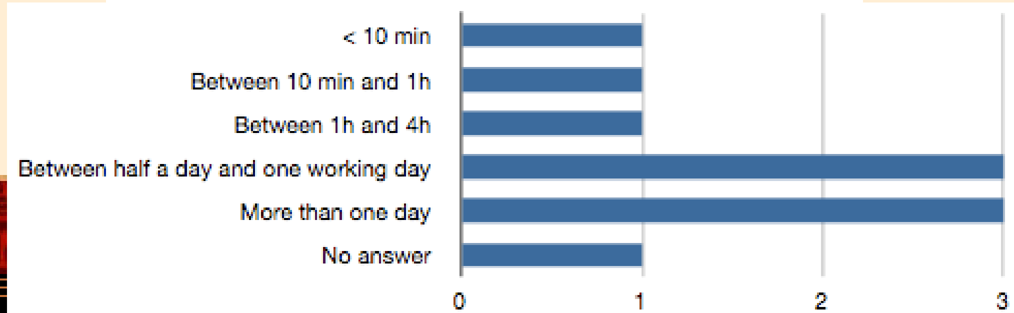


Figure 10: Local / remote preferences (Condorcet method scale)



Users feedback: numerical models (4/5)

- Few community feedbacks:
  - Lack of expertise ?
  - Lack of critical needs ?
  - Lack of visibility on NUM ?
- Collected requirements:
  - Fields solver improvements (e.g. Laplace) (ESA, CETP...)
  - Boundary conditions (e.g. symmetry planes) (ESA, industry)
  - Sources and particles models (multi-species sources...) (all users)
  - Collision models (ESA and industry)
  - Non interacting detectors and advanced monitors (ESA, CETP...)
  - I-V characteristics and scenario improvements
  - Convergence test in the simulation process (e.g $|I_{net}| \leq I_{conv}$)
- General demand on a better documentation and identification
  - Difficulty to select the relevant model and global parameters

Users feedback (5/5)

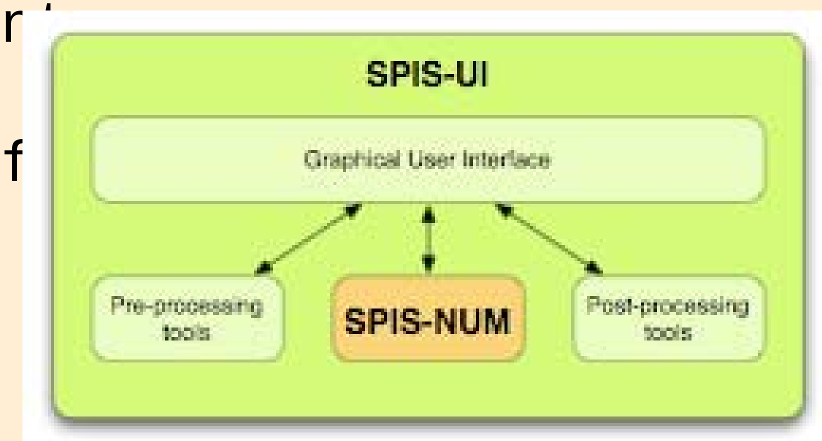
- Main feedbacks from users and future users
  - Easier pre-processing and configuration
  - Improved stability
  - New models
  - Pre-defined scenarios
  - More active community

Outline

- Objectives of the study
- SPINE community
- Users feedback
- SPIS-UI architecture analysis
- Software requirements
- Platform migration
- Community life and business models

SPIS-UI architecture analysis

- Some points to improve
  - Confusing packaging and part of dead codes
  - Imperfect respect of developmen
  - Too intensive use of Jython
  - Data Model and TaskManager re-f
  - Persistency scheme refactoring
  - Global re-factoring needed



- However:
  - Modular and simple design
  - Global respect of canonical design patterns (e.g MVC)
  - Multi-threaded design very well adapted to multi-cores processors
  - Integration of an advanced TaskManager at several UI levels (GUI, batch...)
  - A lot of improvements and cleaning done in the 3.7 RC9 version and since

Outline

- Objectives of the study
- SPINE community
- Users feedback
- SPIS-UI architecture analysis
- Software requirements
- Platform migration
- Community life and business models

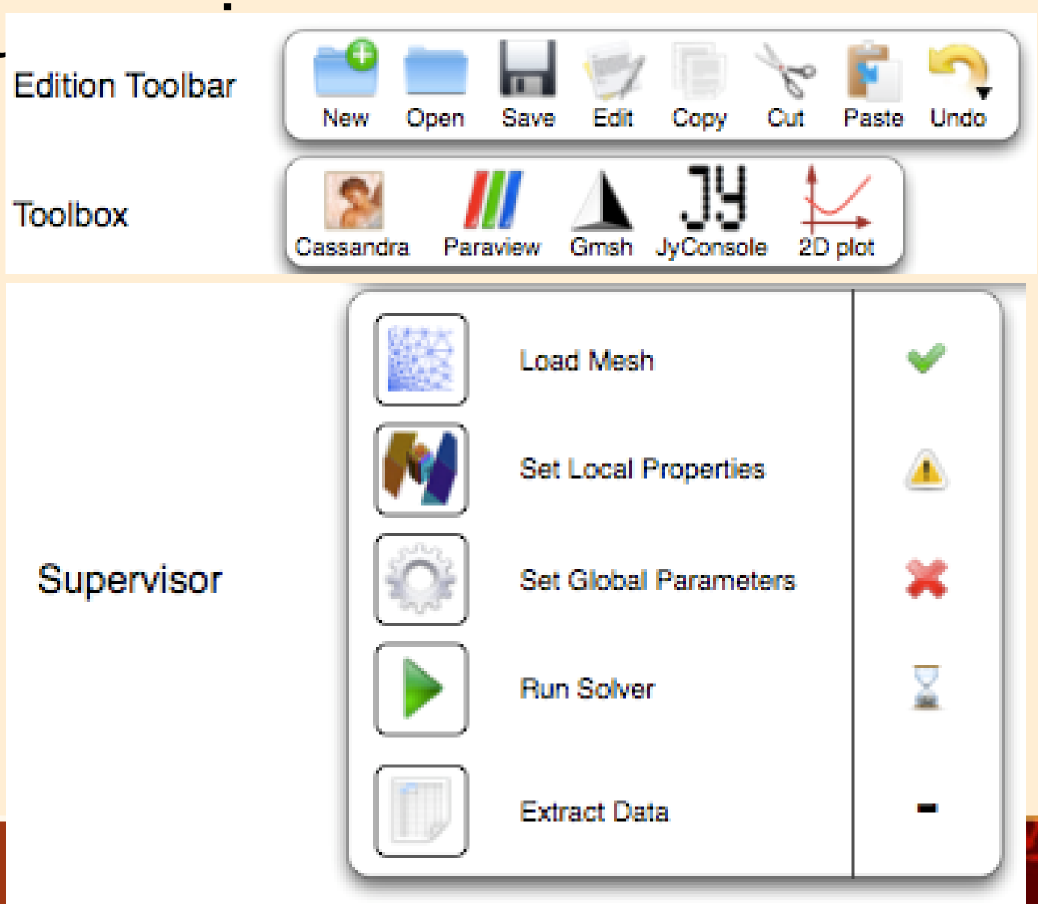
Software requirements: portability (1/8)

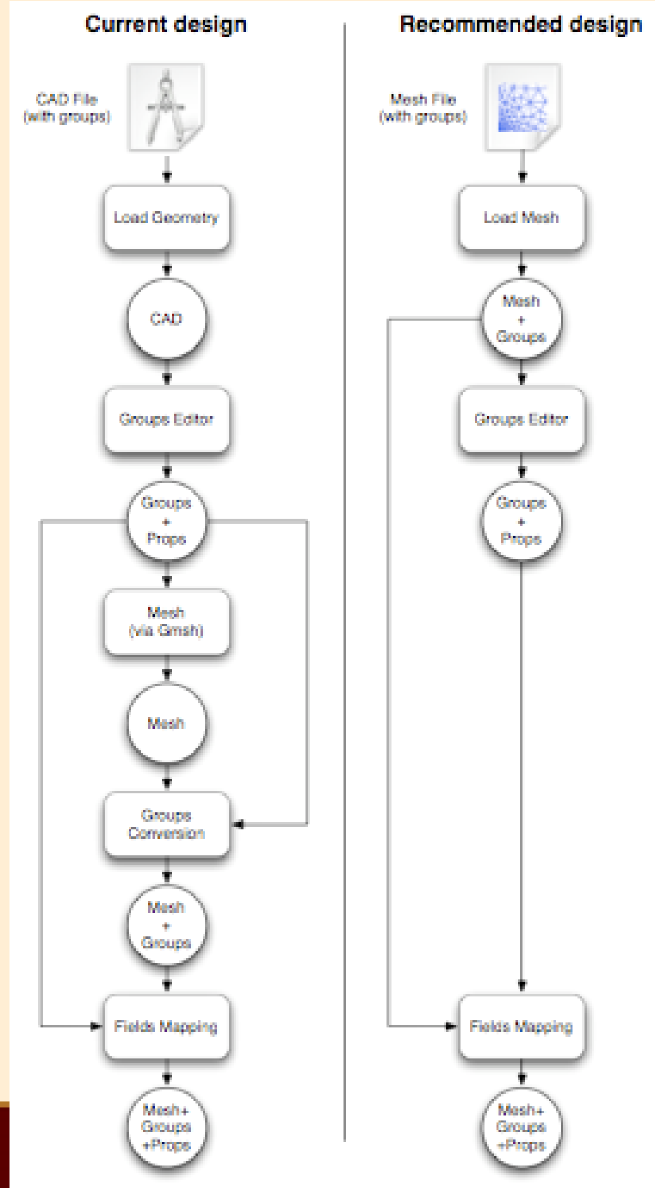
- Highly portable (thanks to the Java approach)
  - Apple OS-X 32/64 bits, G4/G5/Intel Core 2 Duo
  - Linux (Suse, RedHat, Sarge, Ubuntu...), 32/64 bits
  - AMD/Intel Windows 2000/NT/XP, 32 bits
  - AMD/Intel SUN Solaris 10, 32 bits, AMD/Intel

- Remaining OS dependency due to native components (e.g VTK)
  - Possibility to define a “minimal core” without native components and a plugins manager to dynamically load native components
  - Simplified packaging with a distribution for each targeted platform
  - Web-based client deployment

Software requirements: user experience (2/8)

- Modeling chain simplification
- Tool bar simplification and reorganization
- Su





Software requirements: user experience (3/8)

- Project explorer
- DataBus explorer
- Mesh checker
- Improved geometrical modeling manager
- Predefined shapes
- Improved CAD format import => no format appears in the industrial community (STEP is not a widely used yet)
- Improved external CAD tool (other than GMSH)

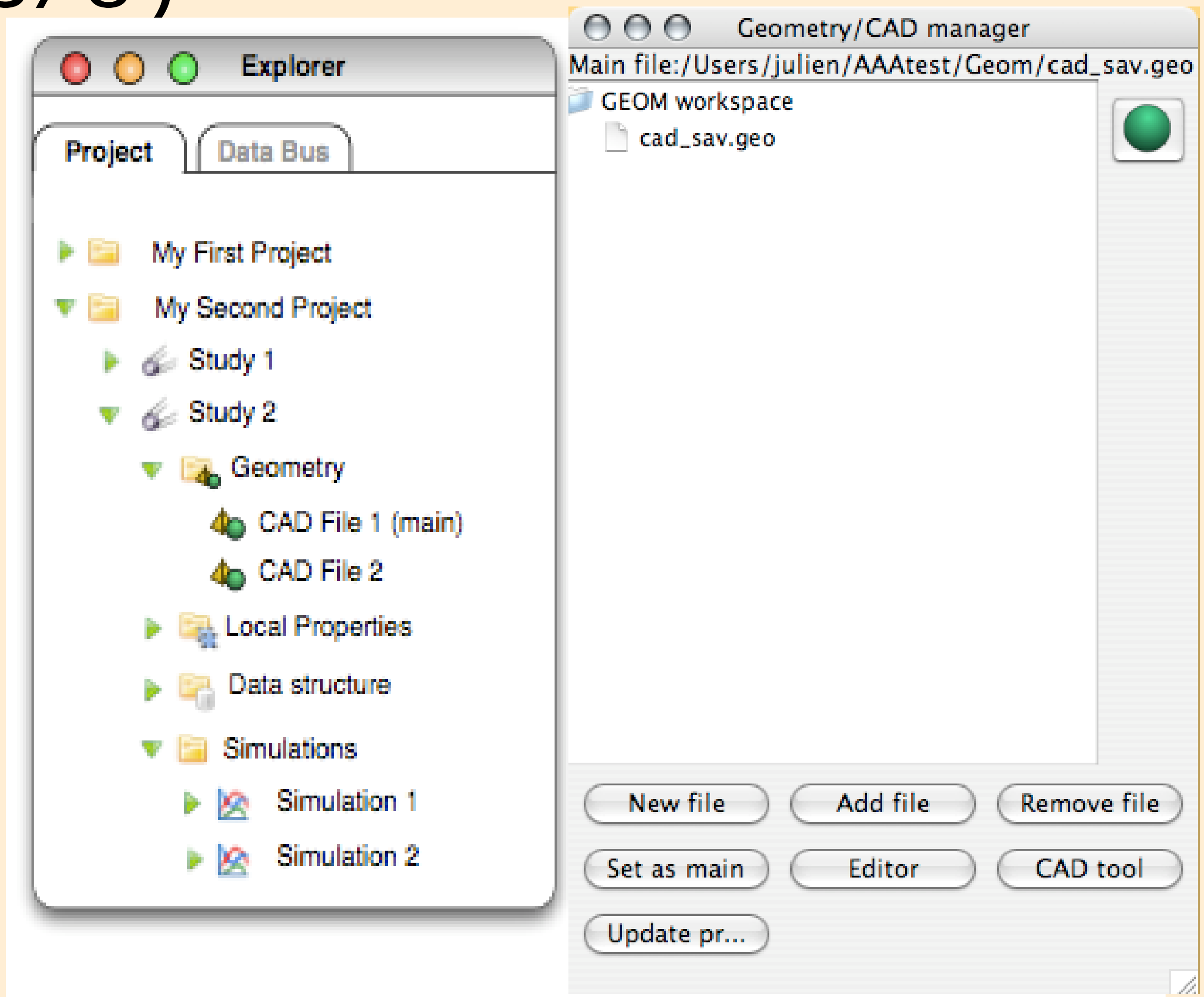
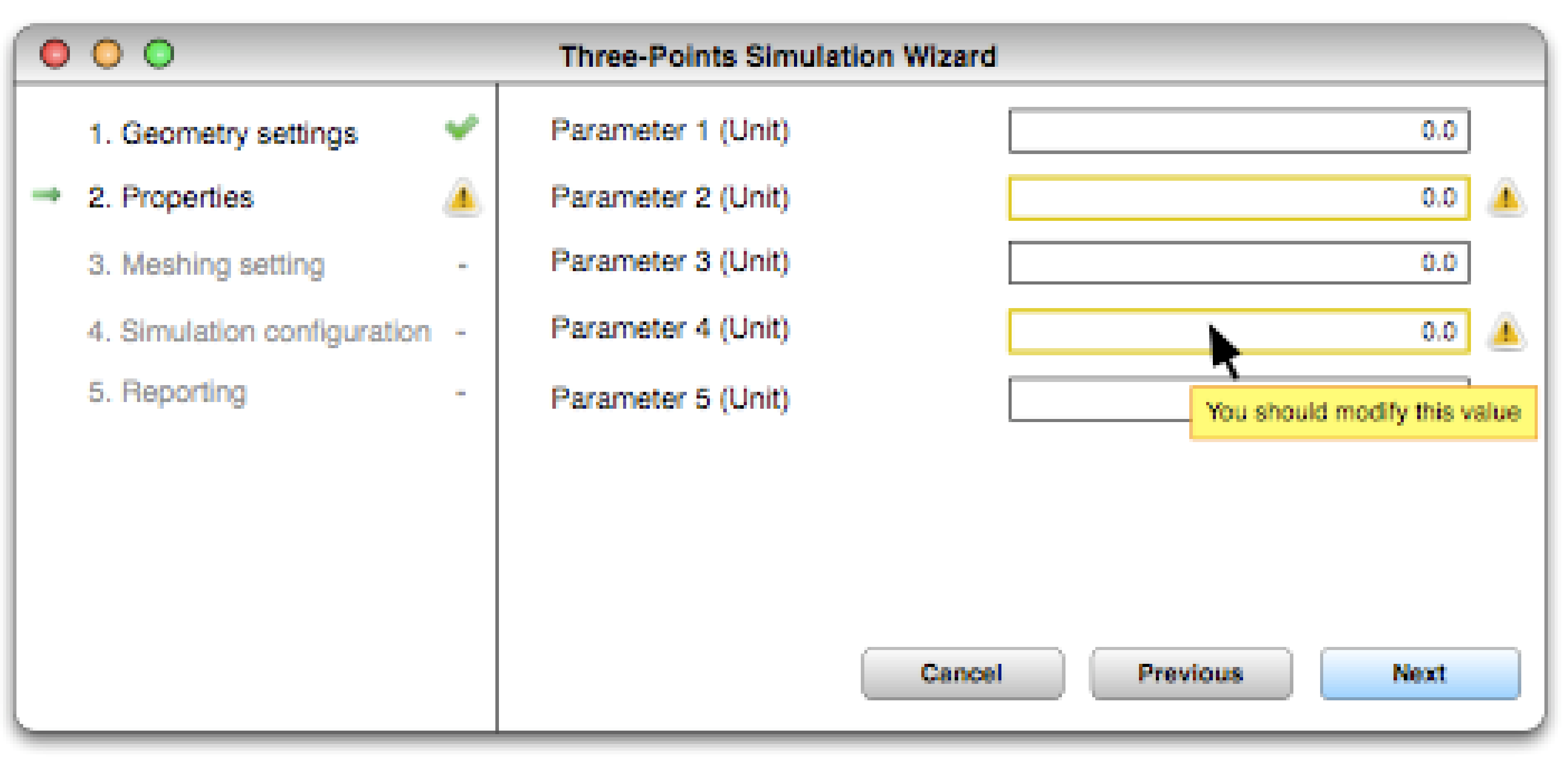


Figure 3: Project and DataBus Explorer illustration

Software requirements: user experience (4/8)

- Guided modeling with step-by-step tailored wizards
  - Worst case and standard cases
  - Levels of modeling complexity



Software requirements: user experience (5/8)

- Live simulation control and monitoring

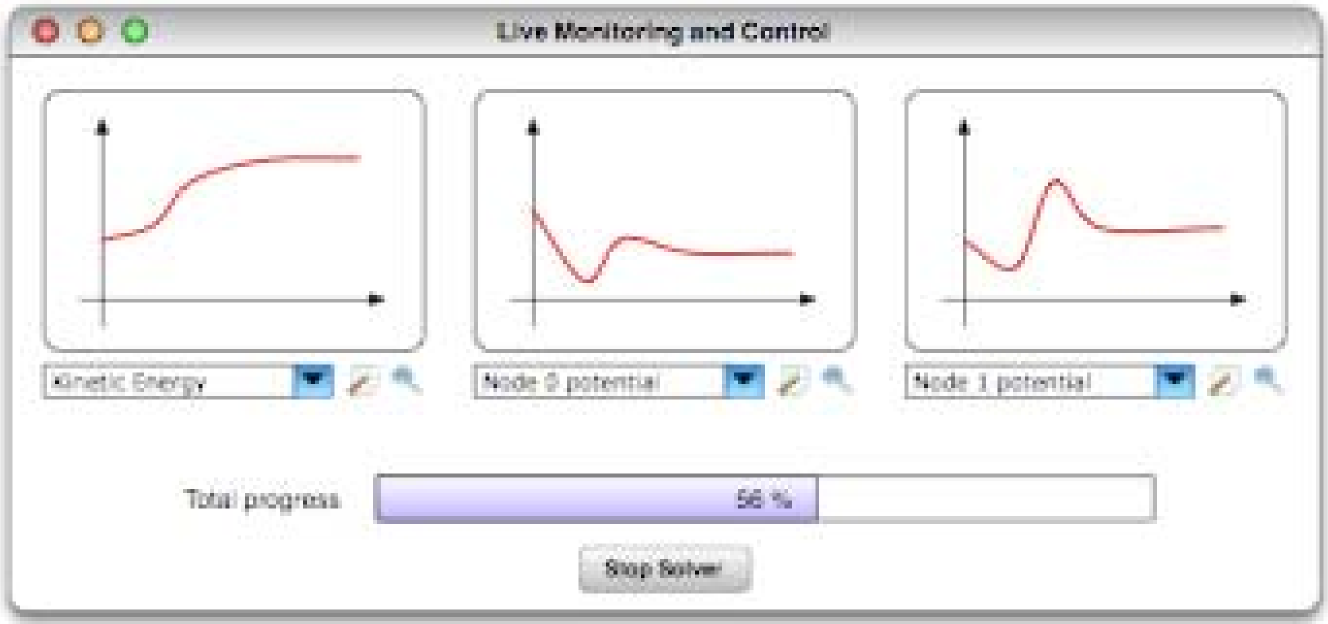


Figure 6: Live Simulation Control and Monitor panel illustration

Software requirements: stability (6/8)

- Improved and common error management
- Code cleaning
- Full Java migration
- Global refactoring

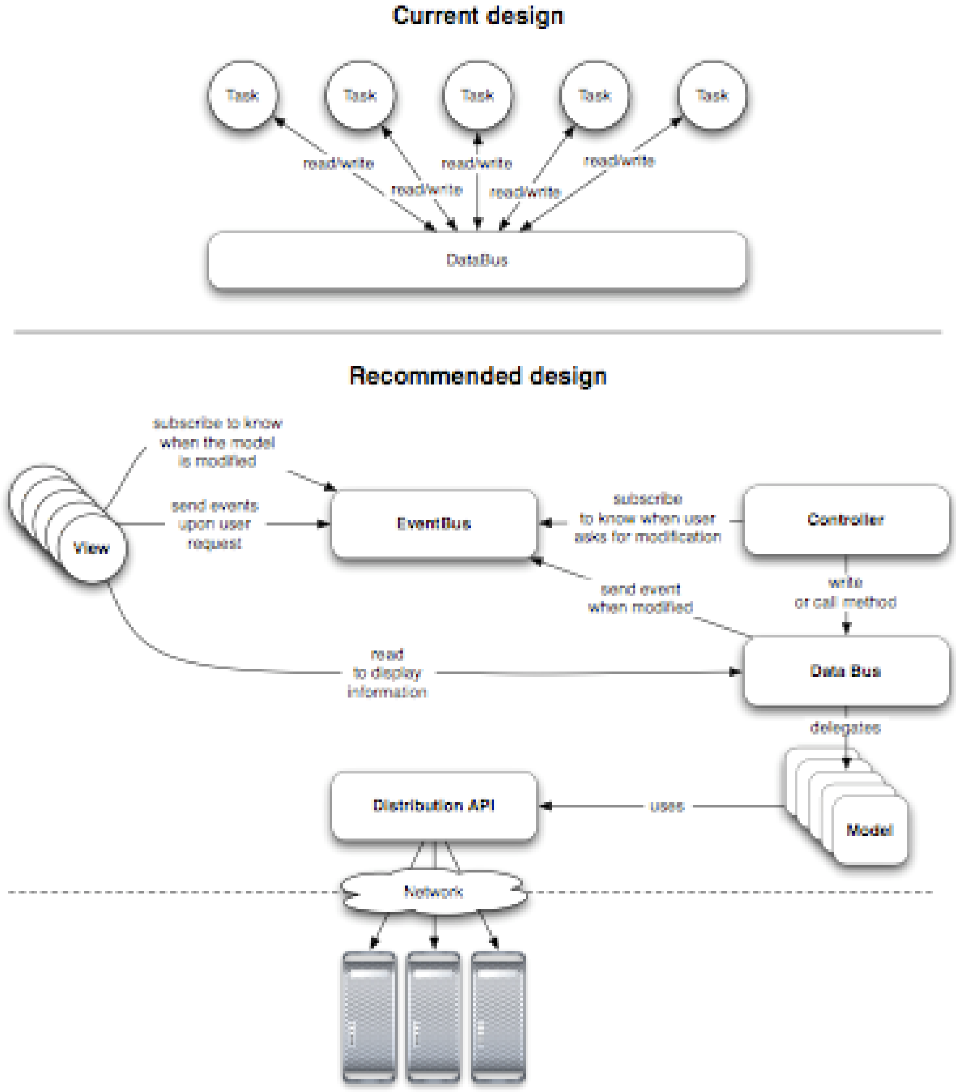


Figure 8: Recommended improvement of data access

Software requirements: Client-Server design (7/8)

- Introduce a distributed design to make benefice of modern clusters and HPC
- Reduce the memory cost at the UI to NUM conversion and address larger systems (grid and number of particles)
- Increase the particles number to improve the statistic
- Computation loop parallelization. Several approaches possible:
  - ProActive library
  - Service Oriented Approach (SOA)
  - Web services
- General evolution in HPC domain

Software requirements: Client-Server design (8/8)

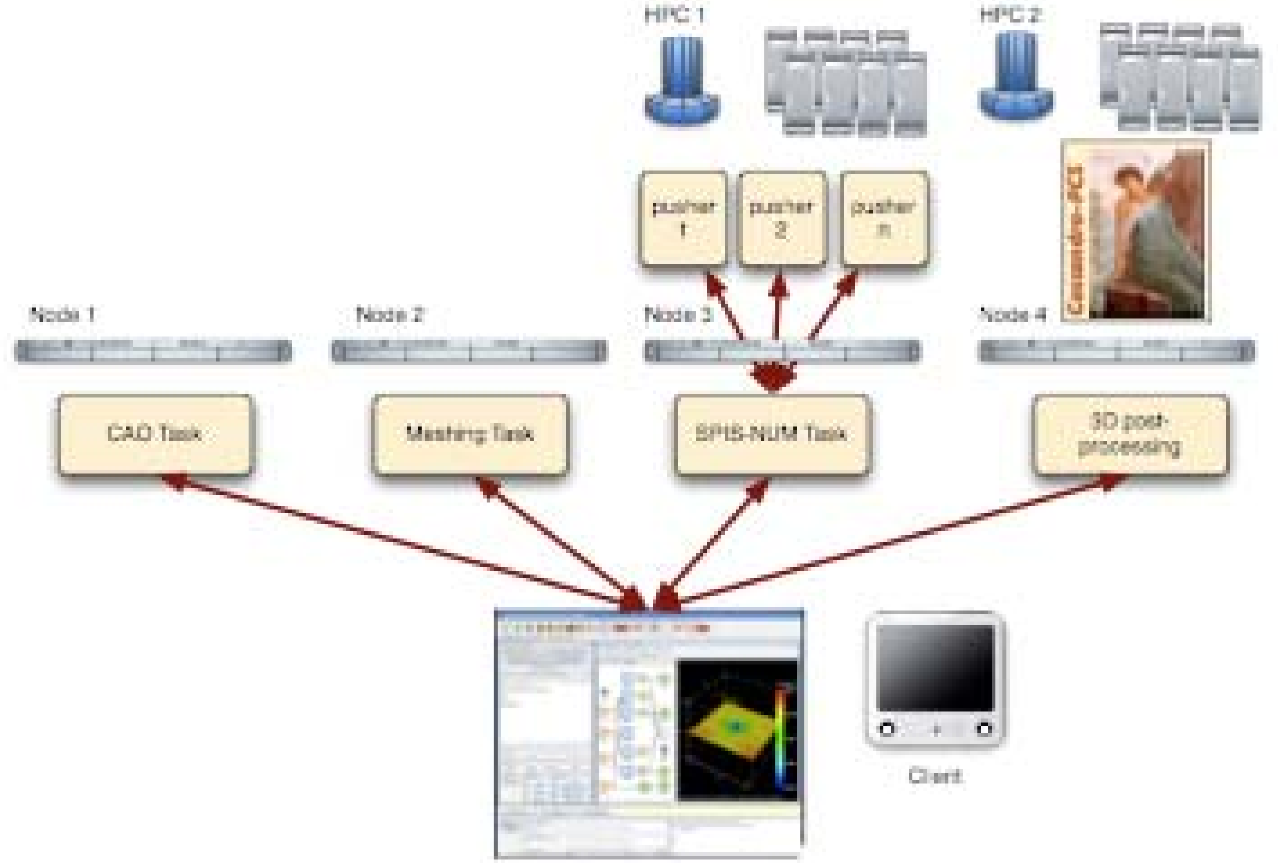


Figure 12 : *Example of distributed architectures, based on the results of experimentation done in the ANR/ SCOS-V3D and ANR/Scorware projects.*

Outline

- Objectives of the study
- SPINE community
- Users feedback
- SPIS-UI architecture analysis
- Software requirements
- Platform migration
- Community life and business models

Platform migration (1/2)

- SPIS is based on a modeling framework or Integrated Modeling Environment (IME).
- Practically, the problematic of a scientific modeling framework outlines several issues:
  - A generic GUI layer
  - A tailored layer or Data Bus, specific to scientific application, to shared scientific data, like meshes, fields and parameters
  - Controlling and supporting external tools
  - Common tailored tools, like 2D/3D viewers and data converters.
- Platform migration should be investigated because:
  - SPIS-UI is mainly supported in the SPINE (small) space related community
  - Currently difficult to maintain (at least for non-experts)
  - New solutions are available

Platform migration (2/2)

- Feedback from other communities
  - IRSN:
    - Developed their own CAD lib based on VTK and CSG description
    - Developed their own Data Model in pure Java
    - Developed their own GUI in pure Java/Swing
  - Scilab:
    - Evaluated Eclipse RCP and Netbean
    - Finally developed their own framework, because better adapted GUI
    - Use components of Keridwen/SPIS-UI (JRosetta, 3D postprocessing)
- Platforms evaluated:
  - Eclipse RCP: too complex, too costly and too risky
  - Netbeans: too complex and too costly (best on the long time range)
  - Keridwen: easy to do, low cost, less risk, but small community (easier to reach in the frame of SPIS-GEO

Outline

- Objectives of the study
- SPINE community
- Users feedback
- SPIS-UI architecture analysis
- Software requirements
- Platform migration
- Community life and business models

Community life and business models (1/5)

- SPIS community exists:
  - Number of downloads
  - Number of scientific papers produced and studies performed with SPIS
  - Activity on the forum
  - Active members present to the SPINE meetings
  - http://www.spis.org is used by the community
- But:
  - Lack of visibility of the SPIS road map by the community
  - Lack of coordination/information exchange about SPIS use and settings
  - Lack of identification of the steering comity/lack of responsibility of members
  - Sub-critical size of a very targeted and space related community
  - Difficulty to maintain the code in an Open Source approach
  - Critical economical equilibrium without the support of agencies
  - Difficulty to introduce the most up-to-date software technologies
- Most of these points have been outlined in:
  - ESA/Open Source Software Study (See Juan Miro)

Community life and business models (2/5)

- SPINE has a real and active life today
  - Numerous members, including from outside the initial community
  - SPIS used by industrial as well as academic
  - SPIS included as reference in several calls to tender
- **SPINE community life and difficulties are consistent with general results of the ESA-OSS Study (R.Ghosh et. al.), United Nations University / UNU-MERIT, regarding scientific open-source projects:**
  - Thematic communities structurally too small (typically about 100 users over the world)
  - Maintenance cost of OSS remains de facto very high (prohibitive in fact)
  - Difficulty to maintain the expertise
  - Difficulty to develop the market, that remains small
  - Difficulty to maintain the long term effort (lack of community life and development road map coordination)
  - Increasing need of quality assurance (validation, calibration)
  - Still need a pro-active and coordinated action of main actors (e.g. agencies, main industrial actors)
  - Only a couple of major actors and few contributions
  - Funding difficulties
- Same conclusion for most of the scientific OSS (e.g Scilab)

Community life and business models (3/5)

- However SPIS and SPINE present several assets and actions compliant with the OSS study recommendations:
  - SPIS begins to be used for spacecraft charging or space related applications:
    - Electrical propulsion
    - Electronic components characterization
  - SPIS hasseveral of its common components mutualised with other projects and outside its initial community and the space related community
    - SPIS-UI/Keridwen is an active part of the ANR/SCOS project
    - Cassandra post-processing engine part of the ANR/SCOS-V3D and ANR/Coll@viz projects
    - JyConsole/JRosetta, the script console, included into Scilab V5 and integrated as standard package into most of the current Linux distribution (RedHat, Debian)
    - Stabilised and growing expertise (large ten of experts in EU)
  - But, still:
    - These elements are not direct effort providers
    - Components mutualisation is more difficult to achieve for numerical models and tailored components
    - Sub-critical economical context
    - Lack of visibility of the community

Community life and business models (4/5)

Recommendations:

- Reinforce the SPINE community structure and coordination
  - An official status for the SPINE (e.g consortium or association)
  - Better clarity and transparency of the SPINE leading
  - Source of funding Scilab consortium model: about 2000€/member/year fee -> about 20k€/year expected => de facto more active members, because
    - Fee payment
    - Official registration (and image identification)
  - Re-introduction of an official steering comity
  - Better clarity on the SPIS evolutions and roadmap
- Reinforced participation of SPINE (as community) to standardization and inter-operability OSS effort
- Better visibility of the services available supports:
  - As OS actions (e.g forums)
  - As commercial services (e.g better identification of services providers on the SPINE platform)
  - Online services (Web services, remote HPC) to bootstrap the use of SPIS and other SPINE tools

Community life and business models (5/5)

Customers and commercial users:

- Industrial users starts to use SPIS or think to use it, but:
  - For most of them spacecraft-plasma modeling remains a sub-critic activity (less than 4 man.month by year)
  - Difficulty to maintain the in-house expertise
  - The easiness to use and the learning curve reduction are critical
  - The funding possibilities are strongly related to
    - Easy access to the expertise
    - Low entry price
    - Easy access to the services (e.g online computing)
  - Requires a better visibility of the services offer
  - Most of them are still in waiting position, i.e "we wait that SPIS is fully developed, validated and stabilized"
- Other projects (e.g Cassandra) have shown that OSS contribution scheme have changed during project
  - No direct contribution because the expertise requirement is too high
  - But private funding for on-demand development, finally reversed to the community in order to mutualise the maintenance cost
  - Requires a better visibility of the software and the road map

Conclusion

- The SPINE community exists today.
- SPIS becomes a real de-facto standard and start to be intensively used
- However, SPIS starts to pay the price of the years
  - Feedbacks of the community and users is very important
  - Evolution of the techniques
  - Contribution possible from other projects
  - Needed re-factoring and evolutions
    - Functional and technical
    - Especially for industrial and engineering applications
- Several technical propositions
  - GUI simplified with a “wizard-oriented” approach
  - Refactoring of the Data Model and better data inter-operability
  - Platform migration
  - Distributed and client-server (including Web based) design
  - Component mutualisation with other communities
- Community life and economical models
  - The SPINE community coordination is a key issue
  - The re-factoring of SPIS, especially for GEO orbits, is a priority.
  - The effort is reachable.

Conclusion

- Do not hesitate to contact us
  - thiebault@artenum.com