SWAPP : building a common framework for NWP-related applications at Météo-France

Éric Sevault, Philippe Marguinaud, Véronique Mathiot, Florence Rabier
CNRM/GMAP and DP/COMPAS, Météo-France
(Toulouse, France)
Warning!

- This talk is somewhere on the frontier of several fields: data management, research and operations interface and methods, visualisation applications, experiments definition, etc...

- ... without being sure that is of valuable interest to any of these fields, taken solely.
Some requirements ...

- An application for recording and retrieving any change in operational suites and resources.

- An application for setting up and monitoring experiments based on operational components. This application should allow duplications and comparisons of experiments, as well as interactive building of new configurations, etc.

- An application for launching and visualizing graphical diagnostics, or at least prepare data input to existing tools (eg. Metview or Synergie).

- A simple archive manager including a garbage collector mechanism.

- An uniform access to these applications for MF people and ALADIN partners (no authorization policy defined yet).
... and some suggestions

- The applications which display the information should be the same that manage and define the information.

- The applications should be self explanatory (as most as possible) in order to improve didactic support to new users of our NWP systems.

- Nothing should be hidden to the user (most of them are advanced users, and their opinion is quite important!).

- Use SMS for the definition of jobs sequence and monitoring.

- Use XCDP as an « external » application as long as a Web-cdp is not integrated in the SWAPP framework.
Yet, an other virtual file system!

- Do not start from the end (the web engine rendering). Of course, we did, because we are always in a hurry to « see » the « nice » graphic interface.

- First version was based on the Baudoin Raoult’s webmars engine, extended to handle multiple applications, users permissions, and some sms features. Integrated in an Apache/mod_perl server. (version 0 in used since 02/2003).

- … and then we started again from scratch, from the bottom, the so-called version 1 (to be delivered 12/2003): what should be the more generic-purpose foundation for any of the application we were supposed to build? Here comes the SWAPP architecture, based on an other VFS.
SWAPP objects

- Within Swapp, objects are persistent: once an object is created by a program, it will still exist (if not removed) when the program exits.
- Swapp objects are Perl objects. They are blessed references in Perl packages which may inherit from each other.
- Swapp objects are shared across processes: each and every program that accesses Swapp databases sees the same set of objects. Programs of various kinds can access Swapp databases; up to now, we have been fiddling with HTTP servers, daemons, shell command line, etc...
- Swapp objects live in a tree whose nodes are objects: each database has a root which has kid objects, which in turn can also have kids, and so on. The database primary model for storing Swapp objects is a filesystem like model.
SWAPP virtual filesystem

- The Swapp virtual object-filesystem makes use of the same approach as the well-known Unix virtual filesystem.
- Its aim is to provide a common interface to all types of object storage (that is to all types of Swapp object databases).
- Swapp databases are analogous to Unix disk devices, and the Swapp Vfs allows to mount those databases onto the root database or an already mounted database. To the end user, the Swapp Vfs just appears as a tree of objects.
- Basic database class is a raw of serialized perl objects in a Berkeley DB. Other kind of databases included indexed database (using DB file again or MySQL) and heavy-attributes database for managing objects with one or several huge components (images, binaries).
Access methods

We have seen that several kind of programs could access swapp objects:

- API: one could write a program using the 'Vfs' main module which defines the main manipulations on swapp objects.

- Psh: more or less the standard shell command line.

- Apache Server (mod_perl), including handlers to SWAPP-based modules. This is the usual user interface to the applications.
User interface : web rendering

- The URL of the HTTP request defines the current application, the object’s path in the VFS, a method to apply and possibly some arguments.

- Basic methods : edit, copy, cut, paste, delete (move to trash), destroy (remove from trash), rename, etc.

- External address for ALADIN partners based on IP identification and firewall http access.
User interface: psh

Add one (or more) new rule in the psh strategy list and then access the full vfs in an interactive way, including node creation and edition.
The OLIVE project

• An effort to share as most as possible common features between operations and research departments.

• An effort to maintain a precise history of changes in main NWP systems configurations in used at Météo-France.

• An effort to document and define any single component of these configurations.

• The wish to have an user-friendly interface for setting new experiments and sharing these experiments between researchers.

• The OLIVE application is now a SWAPP application.
The OLIVE application

- Basic configurations include: ARPEGE France and tropiques forecast & 4DVAR, ALADIN coupling and forecast, toys versions.

- Select a cycle or defaults (in operations at the time of the experiment, current oper or dble).
How to define an experiment?

- Set base hours and dates and any other local setting.

- Cut, paste, edit any task or resource you may need.

- Copy and insert shared resources at any level of the experiment.
Deep in the task...

- A task is an SMS-like object which could run on a target computer. It contains input, run and output sections.

- One section contains one resource which is provided to the section through a method (from oper env, from other xp, from remote file or command, etc.).
The setup objects

- The setup objects could selectively apply to variables of the current family, define a namelist delta, activate or inactivate resources.

- It could be done through selection patterns (such as the base hour).
Refinements

• One could « instantiate » a date or hour.

• Then, all the underlying objects are copied and could be changed independently.
About the OLIVE architecture

- The root experiment node « plays » the SMS definition. The user monitors the experiment with XCDP.

- At any level of the experiment, objects could be « updated » to the SMS server, whatever the current stage of the experiment run is (one could extend the experiment, change the profile of one task, change a resource, etc.).

- SMS fetch the script which is delivered on demand by the sms node object at run-time. There is no « permanent » script at all.
Some graphical diagnostics

- Scores / analyse
- Scores / TEMP
- Obstat
- DDH
- ISP
- Metview plotting
- Retrieve and package data for Synergie
Some OLIVE relatives

• Inside the SWAPP framework, there are some other applications, more or less connected to OLIVE.

• All these applications could be accessed using common interfaces and programs.

• All these applications could share objects which have a common meaning.
The « archive » application

- Mirror of the mass storage archive machine.
- Automatic cleaning policy.
- Interactive cleaning.
- Selection of components for building experiments.
The « GCO » application

- Track history of changes;
- Document components in evaluation stage or in operations;
- Search, diff, etc., on these components;
- Setup next operational cycles contents;
The « history » application

- « Official »
  description of operational changes by the control team.

- 6 models: ARPEGE France, ALADIN France and ARPEGE tropiques (two versions: oper and double suites).
The « subjective control » application

- Some people track strange behaviors of our models on the basis of their own meteorological expertise and produce day-to-day reports, possibly including map snapshots.
Conclusion

- The underlying components (the SWAPP VFS objects and databases) for software development are ready-to-use.
- The genericity and modularity have been sufficient for any application we have been requested to build so far. However, many detailed aspects could be refined, and the SWAPP/OLIVE project is open to any suggestion.
- One important issue is the ability to share experiments or operational settings in the community: therefore the project is very interested in the integration of the PRISM XML «standards».

- Many thanks to Baudoin Raoult, Nils Wedi, Claes Larsson, Pascal Lamboley, Jacques Anquetil and Otto Pesonen.