Pilot Projects in Havana 2003

Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT

NOTE

SAND/01/04

Bjørn Fjellvoll
Jon Gjerde
Knut U. Hollund
Harald H. Soleng

January 16, 2004

Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT

NR Note

Title: Pilot Projects in Havana 2003

Date: January 16, 2004
Year: 2004
Note no.: SAND/01/04

Author: Bjørn Fjellvoll
Jon Gjerde
Knut U. Hollund
Harald H. Soleng

Abstract: In this document we report the results of pilot projects carried out in Havana in 2003.

Keywords: Havana, pilot studies

Target group: Havana sponsors

Availability: Open

Project: Havana license

Project no.: 914002

Research field: Fault modelling

No. of pages: 4

Norwegian Computing Center
Gaustadalléen 23, P.O. Box 114 Blindern, NO-0314 Oslo, Norway
Telephone: +47 2285 2500, telefax: +47 2269 7660, http://www.nr.no

Contents

1 Introduction **1**

2 RMSOpen for communication between RMS and Havana **1**

2.1 Sending transmissibility cubes to RMS 1

2.2 Getting transmissibility cubes from RMS 2

2.3 Communicating horizons with RMS 2

2.4 Communicating faults with RMS 2

2.5 Our recommendation 2

3 Faults in RMS Release 7.0 **2**

4 Variable pillar heights **3**

5 Conclusion **3**

1 Introduction

We have worked with three pilot projects in 2003. These were:

1. Use of RMSOpen in Havana. The aim of this preliminary study was to improve our knowledge about RMSOpen and find out if RMSOpen can be used to achieve better communication between Irap RMS and Havana. Unfortunately, since the current version of RMSOpen (7.0) does not support fault data, we do not recommend using it in conjunction with Havana. This may change with future versions of RMSOpen.

2. RMS 7.0 faults in Havana. The aim of this study was to compare RMS 7.0 faults with simple PFM faults and find out how much work is needed to write a converter. It was concluded that it is best to develop a common data exchange format for Havana and RMS. It is Roxars responsibility to implement this first. It will probably be ready for RMS release 7.3.

3. Variable pillar heights. In order to handle more complex geometries we want Havana to handle pillars that are of different heights. This ability could be introduced in Havana with an estimated implementation time of 8 weeks.

2 RMSOpen for communication between RMS and Havana

Today, the RMS and Havana communicate via files using undocumented file formats. New versions of RMS often change these file formats. This leads to a lot of time-consuming work for the Havana maintainers. In theory, one could use RMSOpen to send data back and forth between RMS and Havana and thus side-stepping the file format problems. A simple test program "TestApl" was written to check if this can work in practice.

RMSOpen is an API (Application Programming Interface) which gives access to the data in a Irap RMS project. It provides access to projects, 3D grids with attributes and wells in a safe manner which Roxar controls. Technically speaking, RMSOpen is a set of C++ classes that can be linked to a stand-alone C++ program (like Havana). RMSOpen promises a method of safe communication between RMS and third-party software like Havana.

For simplicity, the test program "TestApl" is written as a stand-alone program outside Havana. Think of it this way: The program "TestApl" simulates Havana. Everything the test program does, could easily be implemented in Havana. "TestApl" should handle communication with RMS of the three following data types:

1. Transmissibility multipliers cubes.

2. Horizons.

3. Faults.

2.1 Sending transmissibility cubes to RMS

First we want to be able to generate and send cube to RMS. The name of the RMS folder, the name of the transmissibility multipliers cube, the zone, and the realization number where we want to place the cube are specified by the user in a model file. A transmissibility multiplier cube of correct size is made by "TestApl" and then sent to the RMS project. When the task is done, a new parameter can be seen in the specified zone with the name given by the user in the modelfile. This has been implemented and tested in "TestApl".

2.2 Getting transmissibility cubes from RMS

The second task is to get 3D parameters (cubes) from RMS. The name of the RMS folder, the zone number, the realization number and finally the number of the cube we want to get are specified in a modefile. “TestApl” then gets the cube specified in the modelfile and exports the parameter to a file in ECLIPSE format. This has been implemented and tested in “TestApl”.

2.3 Communicating horizons with RMS

Since this case is very similar to the case of sending a cube to RMS, no implementation of code has been done. However, by inspecting the RMSOpen user manual (Release 7.0), we see that this should work just as for the cubes.

2.4 Communicating faults with RMS

Faults are unfortunately not yet supported in RMSOpen and Roxar has apparently no plans to add this functionality. If external funding is found, Roxar thinks such a job should be a cooperation project between Roxar and NR.

We have asked Roxar if they see any special implementation problems and what the probable time frame for such a project would be. Our impressions from these conversations ae that it should be relatively easy to get fault data from RMS, but to append fault data to RMS would require a bit more work, especially if RMS Y-faults should be supported. To add support for faults (except Y-faults) would probably take a couple of weeks. Y-faults would take much longer.

2.5 Our recommendation

Because faults are not supported in the latest version of RMSOpen (release 7.0), the use of RMSOpen to improve communication between RMS and Havana is limited. However, we can send horizons and cubes to RMS and also get horizons and cubes from RMS without using files as communication method. But the true potential in achieving better communication, cannot be obtained before RMSOpen supports faults.

In view of the above we recommend using common file formats for data exchange.

3 Faults in RMS Release 7.0

The aim of this study was to compare RMS 7.0 faults with simple PFM faults. We wanted to estimate how much time it would take to write export/import functions between these two fault formats in Havana. At the same time we wanted to learn more about RMS 7.0 faults in general.

The RMS fault surface object consists of many fault surfaces with truncation information between these. It contains the relevant information, but unfortunately it has no export function. It can be saved to an internal RMS format. This internal file format is a binary dump of the C++ class structure of RMS and hence prone to frequent changes as base classes deep in the code structure are modified. The format is undocumented and a lot more complicated than other data formats that Havana handles.

The fault communication problem was discussed with Roxar in a meeting. We made the following decisions:

1. We should create a common data exchange format for Havana and RMS.
2. It is Roxar's responsibility to implement this first. It will probably be ready for RMS Release 7.3.

Individiual RMS faults are defined with splines and are part of a network structure. PFM faults are pillar based and possess an invertible faulting operator. Hence, even if the file format problem were solved it would not be trivial to convert between the two. Nevertheless, we would be able to deliver Havana support for the new exchange format as soon as RMS begins to supports it.

To implement conversion to and from this new format, will take approximately four weeks.

4 Variable pillar heights

The parametric fault modelling (PFM) format has the limitation that all pillars are scaled to the same vertical length. This is done internally in Havana in order to reduce the mathematical complexity of some calculations. Unfortunately, the price to be paid is a significantly reduced flexibility.

In practice we face the problem of crossing pillars and invalid pfm faults when converting from other formats using pfm2pfmEclipse or Ecl2pfm (Eclipse to PFM converter). We also foresee problems with future implementations of Petrel2pfm (converter from Petrel fault files to PFM) or RMS2pfm (converter from RMS faults to PFM). In order to handle more complex geometries and to avoid present and future problems we need to relax the internal assumption that all pillars are of equal height.

The pilot project has established that this extension is feasible. The necessary algorithms have been developed and some fundamental algebraic equation solvers (polynomial equations up to an including fourth degree) have already been implemented and tested in Havana 5.2. We have located the code lines that need to be changed in the modules:

- PFM
- Pfm2PfmEclipse
- Eclipse2Pfm
- InputParametricFaults
- SimulatePFM
- StructuralModelling

Changing the internal fault representation is a fundamental change in core elements of the parametric fault modelling modules. This is not a task to be underestimated. To secure a high quality result it is therefore very important that each step of the new algorithms are thoroughly tested and validated before this reimplementation of PFM is shipped. We estimate that no less than eight weeks are needed to implement and test the new modules.

5 Conclusion

RMSOpen has been evaluated as a vehicle of communication between Havana and RMS. Due to limitations in RMSOpen we recommend to stick with file communication but to try to establish common standards for the file formats.

The RMS 7.0 faults have been compared with simple PFM faults and work has been initiated on a common exchange file format for faults. A full implementation in Havana takes about four weeks.

Difficulties in handling complex fault geometries in Havana has been traced back to inherent simplifications made in the internal handling of PFM formats, i.e., the homogenization of pillar heights made in Havana. A preliminary study on variable pillar heights has been carried out and necessary new algorithms have been developed and tested. A full implementation of this improvement will take about eight weeks.