A Portfolio Model for Natural Catastrophe Reinsurance

Experience using Ada and the GNAT programming environment

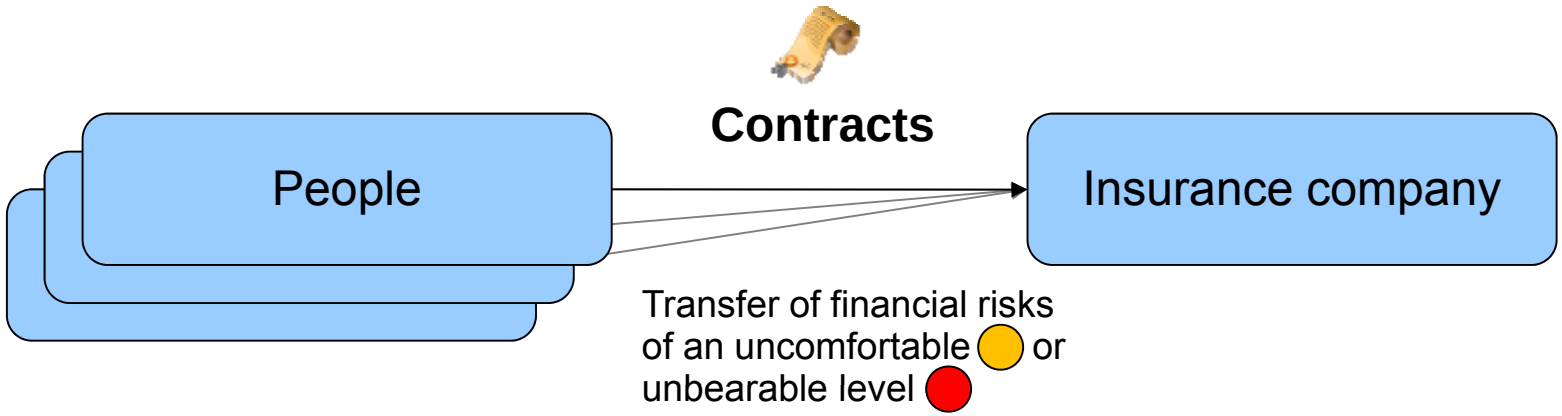
Dr Gautier de Montmollin

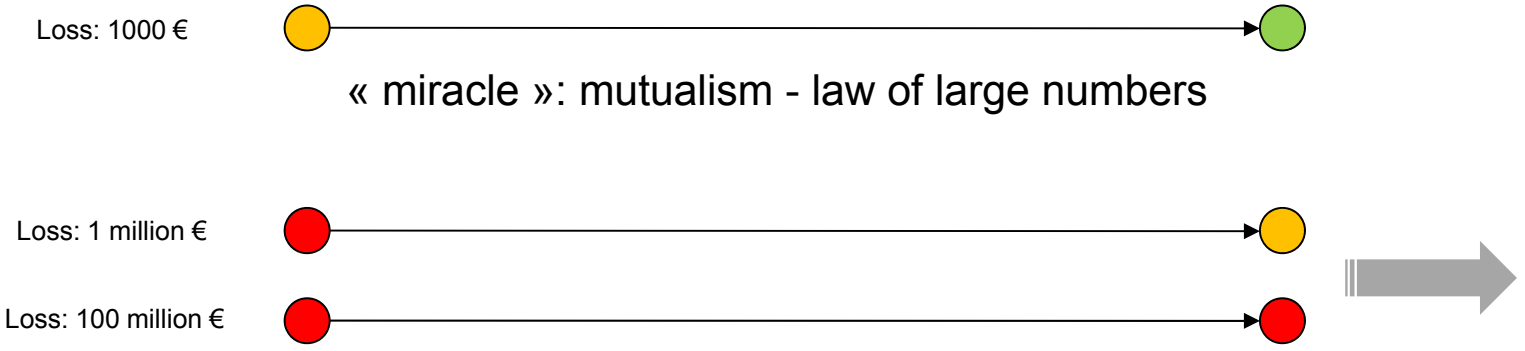
Portfolio Analyst, Business Unit Catastrophe, Zurich

Disclaimer

The following presentation is for general information, education and discussion purposes only, in connection with the Ada-Europe 2012 Conference. Any views or opinions expressed, whether oral or in writing are those of the speaker alone. They do not constitute legal or professional advice; and do not necessarily reflect, in whole or in part, any corporate position, opinion or view of PartnerRe, or its affiliates, or a corporate endorsement, position or preference with respect to any issue or area covered in the presentation.

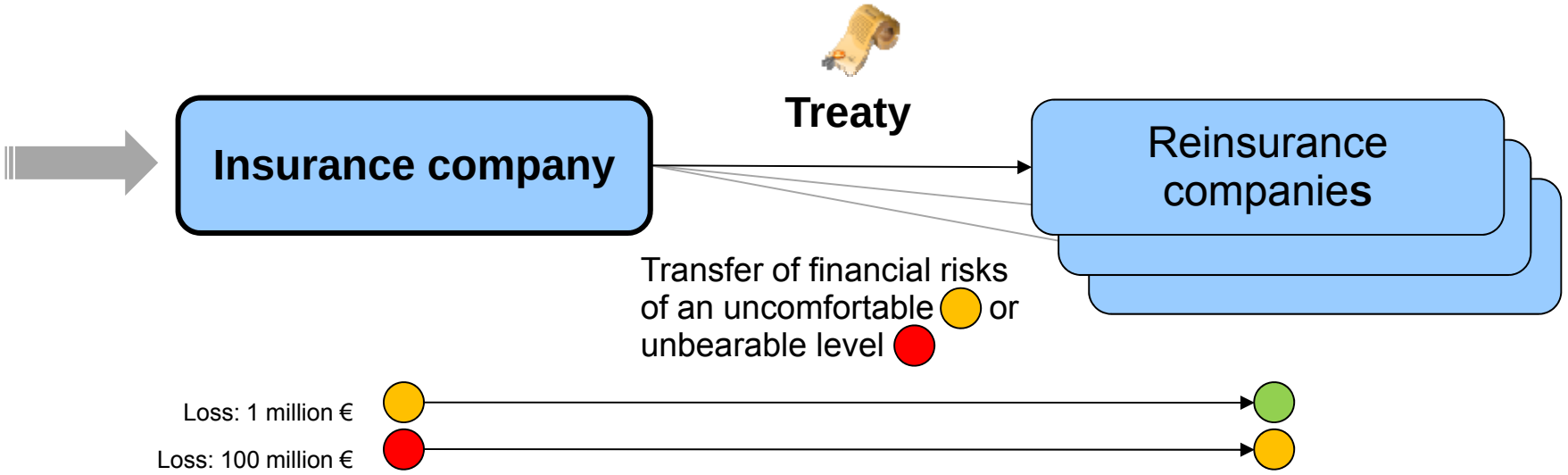
What is reinsurance ?



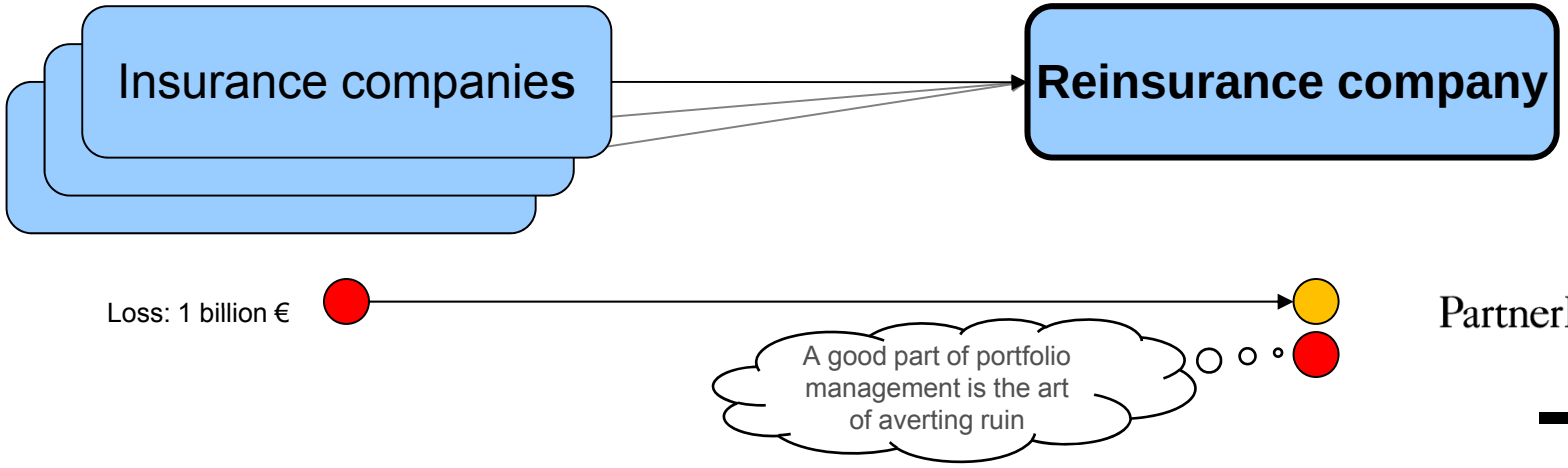


PartnerRe

What is reinsurance ? (continued)



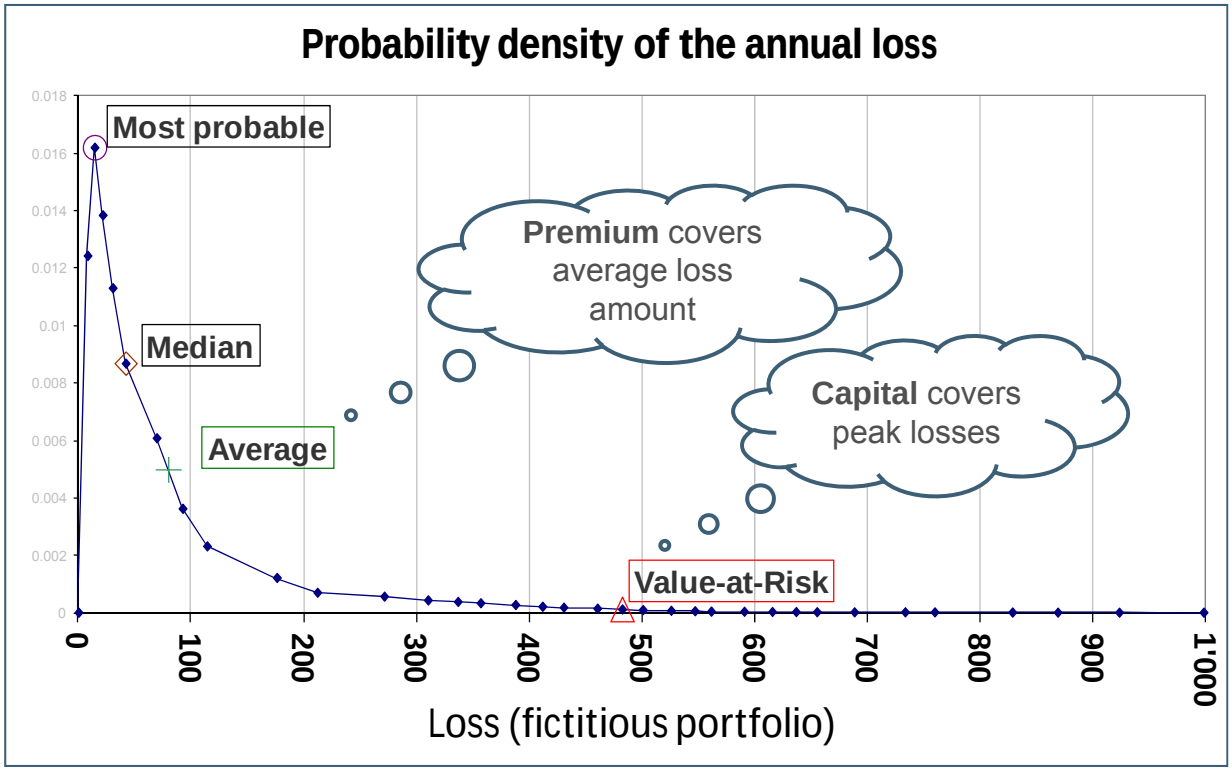
Portfolio of treaty shares



PartnerRe

Use of Portfolio Modelling

1. Compute how much **risk-based capital** is needed.



Worldwide Catastrophe loss distribution

PartnerRe

Use of Portfolio Modelling (continued)

2. Quantify **diversification** and **profitability** over regions and types of perils (~400 peril zones)
3. Consider dependencies

→ **Large probabilistic model**

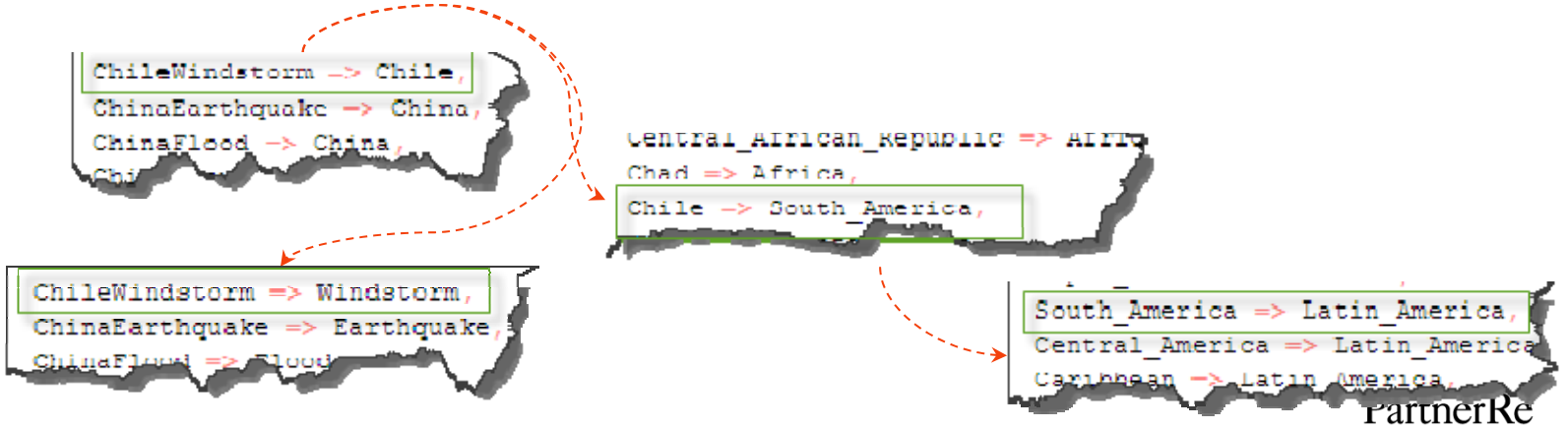
Active use, beyond reporting: **portfolio planning and optimization.** Goal: reduce the capital at risk while keeping a certain volume of business and profitability

How did portfolio optimization cross Lady Ada's way ?

- Portfolio optimization initiative started 2007
- **Idea: genetic algorithm** with random mutations (recent addition: selection) of a portfolio
- **Problem:** the software available in 2007 could not be used for that: 1/2 hour to load a portfolio, > 1/2 hour to compute a single portfolio
- Fall-back, at that time, was to consider over-simplified portfolios, or script an external software (slow *and* over-simplified) 
- **Solution**…

Portfolio optimization & Lady Ada (continued)

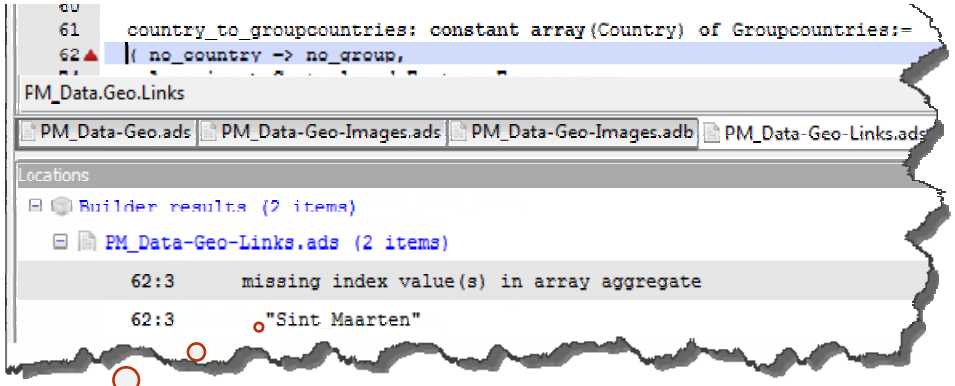
- **Solution:** write a new model from scratch with run-time performance *and* easy maintenance in mind
- **Precomputation**. Furthermore: move **run-time** effort to **compile-time** effort. Embed **constant data** (esp. geography).
- From database to source code: enumerated types (type PerilZone: more than 4000 items!) and arrays with initialized data in an arbitrary, human-readable order:



Data embedded in source code

- Maintenance of data as source **without** Ada: would be more difficult than in a database, or would need to be at least backed by a database (SQL scripts to regenerate sources)
- Maintenance **with** Ada: easier than in a database. Source code generation had to be done only once, in 2007.

Real example of maintenance change:



New country, from the ex-NL Antilles

PartnerRe

Data embedded in source code (continued)

Gain:

- performance: thousands-fold *), portfolio is loaded within a second
- code safety (language rules on arrays and case statements)
- data safety: any error in data is caught – would pass through, in SQL

The number-crunching side…

- Monte-Carlo simulation to compute each portfolio
- Matrix computations for getting correlations into random vectors
- Choice of platform: native; and compiler: GNAT, with optimized machine code generation

Gain in performance: 20x*) : a whole portfolio is computed in < 2 minutes

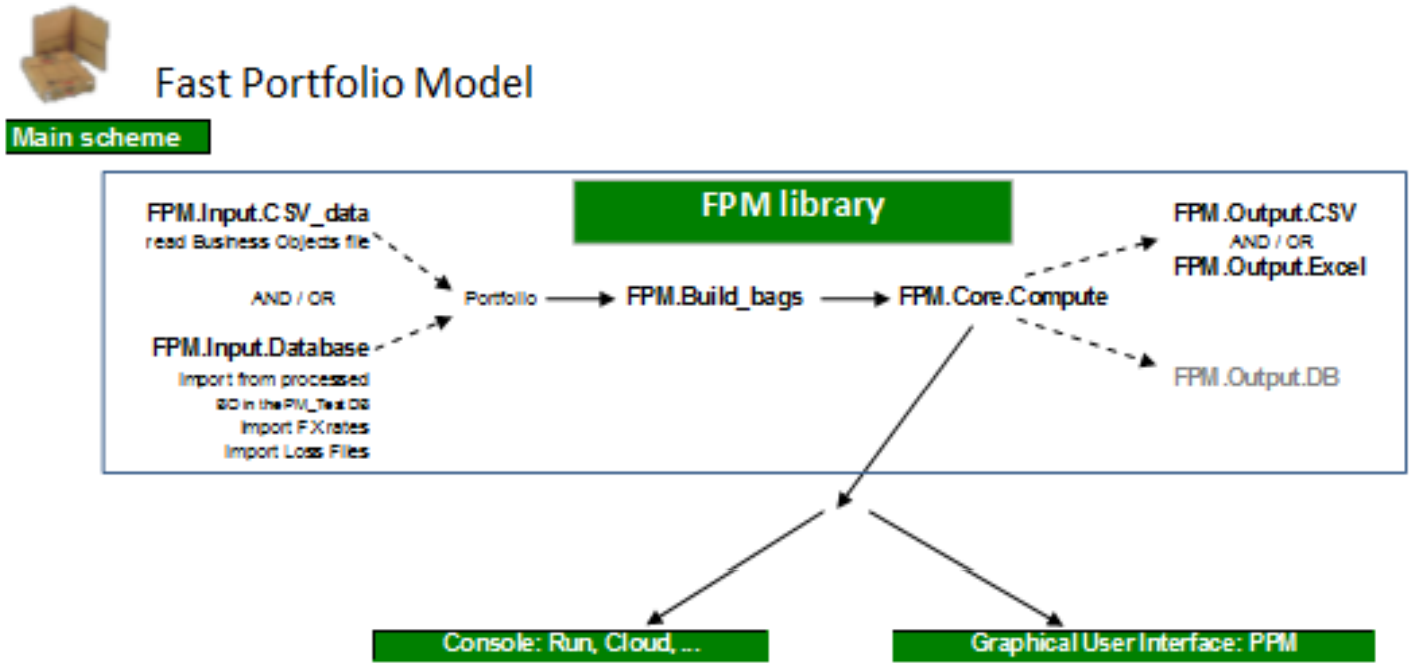
- Possibility of not using access types (pointers, reference types) !

→ faster **and** safer – unusual with IT!

PartnerRe

*) Compared to the previous generation of the Portfolio Model

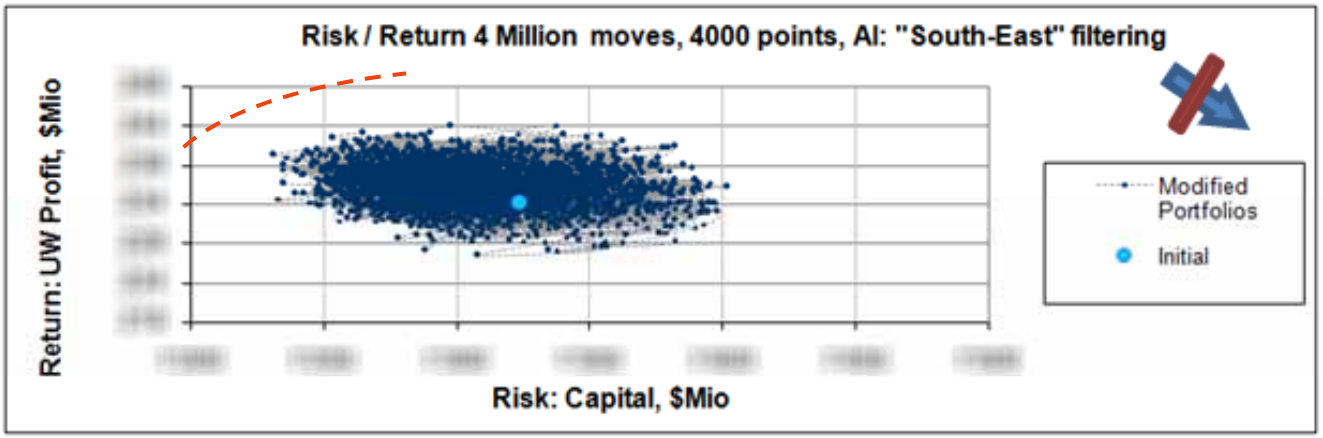
Fast Portfolio Model architecture



type Portfolio_data → type Portfolio

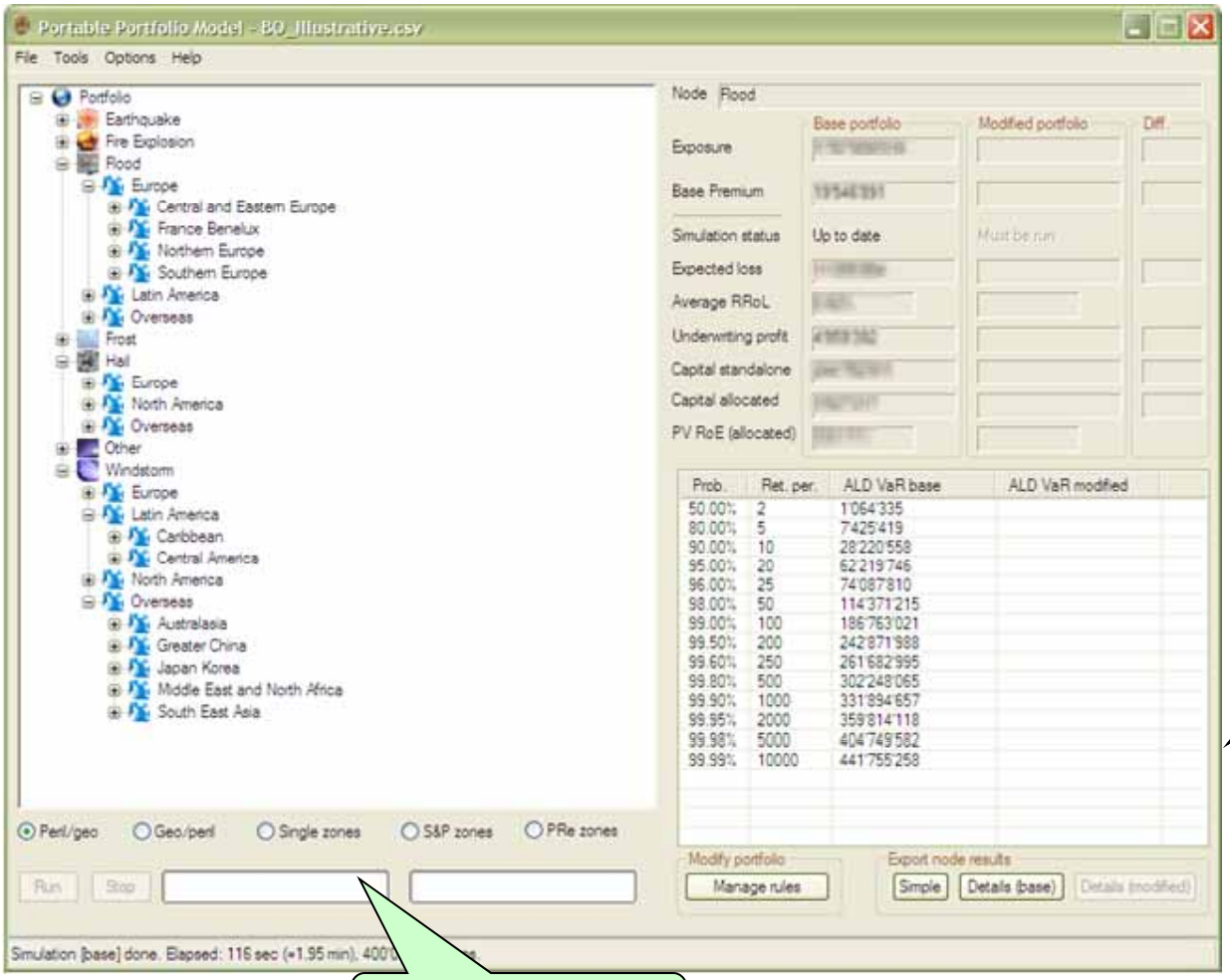
PartnerRe

Out of the factory: Portfolio Optimizer



- **Mutation** of a portfolio by moving exposure from a peril zone to another one, all by rescaling treaty shares (must stay realistic)
- **Selection** rule: reject variants with more risk and less profit

Out of the factory: the Portable Portfolio Model



100% made with free, open-source components: GNAT GPL, GWindows, Excel Writer, database lib, math libs…

Multicore

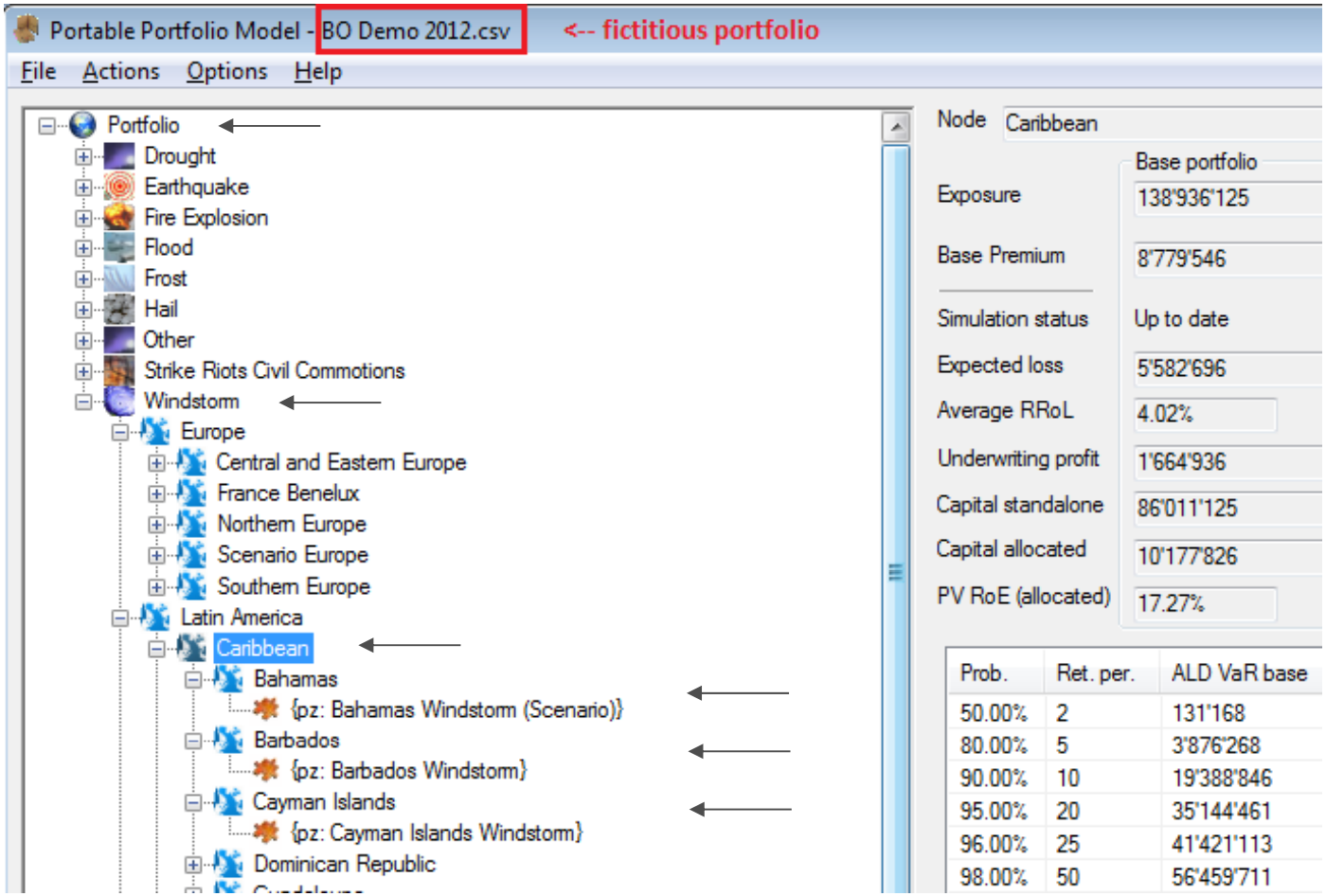
PartnerRe

Portable Portfolio Model (continued)

```
package PPM_Main.Daemons is

  task Status_display is
    entry Start;
    entry Display(pw: in out PPM_Main.Main_window_type);
    entry Stop;
  end;

  task type Simulation_type is
    entry Start;
    entry Run(pw: in out PPM_Main.Main_window_type; vers: PF_Version);
    entry Stop;
  end;

  Simulation: array(PPM_Main.PF_version) of Simulation_type;

end PPM_Main.Daemons;
```

PartnerRe

Portable Portfolio Model (continued)



PartnerRe


PartnerRe