

# PillarOne – Life Modelling Software



Stefan Zumsteg

Intuitive Collaboration



- Designed for enterprise use, data base oriented
- Native support of all major database formats.  
Not limited to data import:  
Results are written to a database as well

## **Advantages:**

- All the well known advantages of professionally managed data handling: reliability, backups, ...  
...historized save of all changes -> audit trail
- Exploratory data analysis of results beyond predefined standard outputs

PillarOne uses dedicated structured data types for *all* business objects, like accounts, persons, policies.

An account for example is more than a collection of figures in cells.  
It has

- a currency
- a balance (of course...)
- a list of bookings  
(each with currency, amount, sender, receiver, transaction date)
- rules how and when interests are accrued

Structured data types provide additional safety.

Structured data types give you additional power

### Structured data types prevent things like...

- ... looking up the wrong value of a mortality table
- ... adding nominal amounts of different currencies
- ... discounting amounts in € with A US\$ yield curve
- ... changes in FX or interest rates not affecting *all* items linked to that currency

### **Account** (data structure)

An account is more than just a balance figure.

It has informations about its owner and maintains a list of transactions, with sender, receiver, amount and date.

It's data can be processed in various ways , e.g. to get

- accounting figures  
(premiums booked between ... and ...),
- economic figures (present valued of cash flows with respect to a given date)
- both during simulation and afterwards.

It can be used as a foundation to define a reinsurance contract strictly on accounting transactions ...  
... just like the real deal.

```
3  import ...
8
9  /**
10 * @author stefan (dot) zumsteg (at) intuitive-collaboration (dot) com
11 *      Date: 12.02.2009, 14:15:07
12 */
13 public class Account {
14     public String accountID;
15     Currency currency;
16     double balance;
17     protected List<CashFlow> movements;
18
19     public Account(String accountID, Currency currency){
20         this.accountID=accountID;
21         this.currency=currency;
22         balance=0;
23         movements = new ArrayList<CashFlow>();
24     }
25
26     public boolean book(CashFlow cf){
27         if (cf.currency!=currency) return false;
28         if (cf.sender==this) {balance-=cf.amount;
29             movements.add(cf);
30             return true;}
31         if (cf.beneficiary==this) {balance+=cf.amount;
32             movements.add(cf);
33             return true;}
34         return false;
35     }
36
37     public double getBalance(){return balance;}
38     public Currency getCurrency(){return currency;}
39 }
```

## Which elements *may* be modeled stochastically?

### **Persons**

- Death
- Disability
- Lapse

### **Assets**

#### Prices

(almost) any desired level of complexity is conceivable