

Simulation of Asset/Liability-Profiles as Part of an Insurance Management Simulation Game

Peter Schenk

Abstract

The high complexity of the problems to be solved within the domain of asset/liability-management induces a demand for models and their software implementations in order to simulate asset/liability-profiles. This report introduces the prototype of an object oriented insurance management simulation game which bears the following features:

- it generates and visualises effects of asset/liability-management for training purposes,
- it contains the potential of being adapted to real situations to serve as a tool for management decisions.

Résumé

La grande complexité des problèmes à résoudre par des entreprises d'assurance dans le cadre de la gestion des "asset/liability" a généré la demande de modèles complétés par un software adéquat, grâce auxquels il est possible de simuler des profils "asset/liability". Cet article introduit le prototype d'un jeu d'entreprise d'assurance conçu en tenant compte des caractéristiques suivants:

- les effets de la gestion des "asset/liability" sont générés et rendus visibles pour des buts de management training,
- en même temps, grâce à sa structure modulaire, ce jeu peut servir de point de départ pour des décisions réelles.

Keywords

Asset/Liability-management, management game, small talk, capital market model

1. Introduction

What could be the motives and benefits of integrating A/L management facilities into an insurance management simulation game? The following article answers this question showing the results of interdisciplinary research. In its course it turned out that the desirable features the implementation of a A/L model should have happened to be readily available in the concept of Object-VersPlan, an object oriented simulation game implemented by Ralf Klotzbücher. The next chapters deal with the complexity of A/L management, the design of Object-VersPlan and how a synthesis could look like. The results shall be illustrated by an example. In the final chapter comments on the potential and the limitations of this approach can be found.

2. The Asset/Liability Model

The big task of A/L-Management is to manage the overall risk situation of a company. The usual approach is based on the concept introduced to finance theory by Markowitz and Sharpe: market values of all the assets and liabilities are aggregated and the risk of the resulting quantity is measured. At first sight this suggests a bottom-up approach as follows:

- Divide the set of assets and liabilities into homogenous subsets; do this until you identify portfolios the future market value of which can directly be prognosticated¹. The smallest of these portfolios will consist of single assets or liabilities (e.g. equities, insurance policies), and the prognosis of their market values will possibly be a sequence of probability distributions, i.e. stochastic processes.
- Identify the causal connections between the thus constructed sets and subsets; thus a risk hierarchy is derived wherein the vertices² denote sets of assets and liabilities and the edges show cause/effect relations.

- Determine the functional relations between "effect" and "cause" vertices and calculate the market values of all sets starting at the leaves until you arrive at the prognosis of the overall market value.
- Determine the overall risk position according to a predefined risk measure, e.g. duration (of both assets and liabilities) or standard deviation (or volatility according to the usual terminology in finance theory).

This approach means the complexity of the whole problem is reduced step by step by pooling risks that show a direct cause/effect relationship until all vertices are represented by one final set (set V8 in the below diagram). The resulting sets of vertices shall be called *vertical portfolios*.

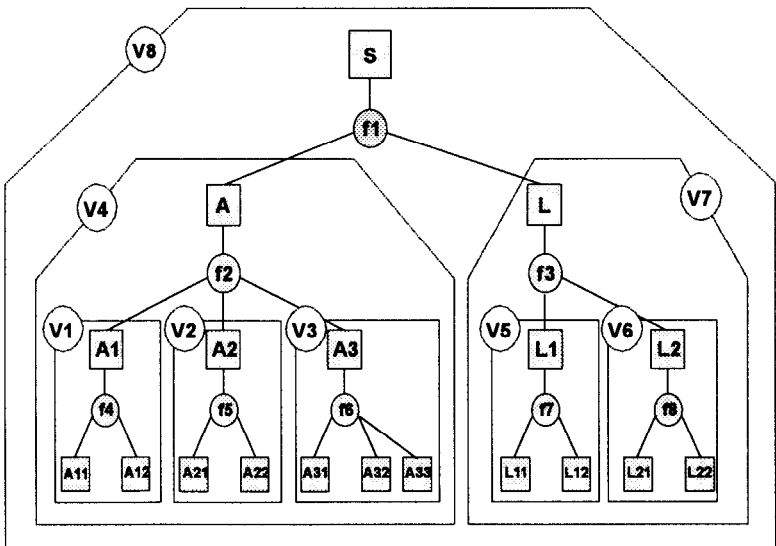


Diagram 1: Construction of vertical portfolios

Obviously the risk manager who follows this approach is confronted with various problems. He needs a complete collection of input data and relations and, as usually an analytic solution will not be possible, has to aggregate them by means of stochastic simulation.

In order to simplify the complexity of the process of aggregation several methods can be thought of. One is to reduce the stochastic problem to a deterministic one and take the uncertain part of it into account by calculating scenarios. Another possibility is to use information on the current situation of the specific company in order to search an improved sequence of aggregating vertices. How can this be done?

Usually additional information on existing correlations between vertices that are not neighbours in the risk hierarchy is available. This information shall be called *expert knowledge* and can be used to reduce the complexity of the risk hierarchy by pooling the corresponding vertices in sets that shall be called hereafter *horizontal portfolios*. This expression allows for a clear distinction from the above vertical portfolios.

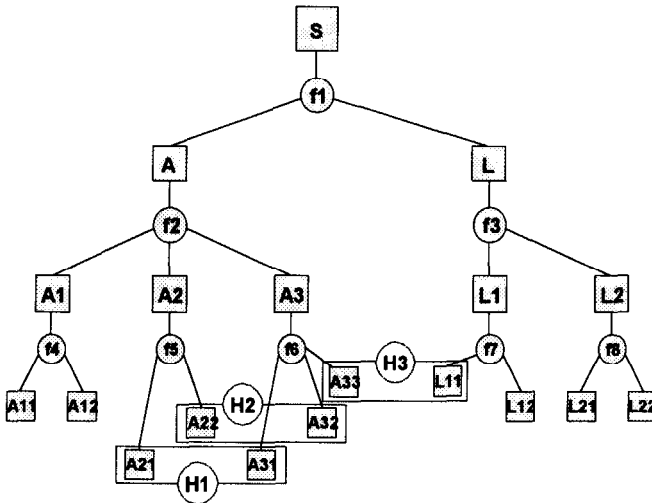


Diagram 2: Construction of horizontal portfolios

An admittedly rather trivial example would be to directly prognosticate the market value of a portfolio consisting of an option and its underlying instead of adding the individual forecasts on option and underlying (this could result in portfolios like portfolio H1 and H2 in the above diagram). Another source of expert knowledge is the information contained in the valuation of a product. Here information on the correlation between premiums and claims is contained. Therefore, instead of separately simulating a liability and its corresponding assets it might lead to more reliable forecasts to directly simulate

their balance (e.g. the cash-flow coming from portfolio H3 in the above diagram). An obvious example for this approach is unit-linked policies. A change in the market value of the asset consisting of the units related to a policy simultaneously has the identical effect on the corresponding liability. If furthermore the actuary has the right to review premiums in case the value of existing mortality risk premium is insufficient to support the cover, the risk of having to finance a possible mortality loss when market value of units is less than the minimum sum assured can be at least theoretically eliminated. Thus taken on its own the liability is subject to a investment risk and a mortality risk and the asset on its part is subject to an investment risk. Combining the expert knowledge about the liability and the corresponding asset the investment risk is eliminated and the mortality risk significantly reduced or even eliminated.

It can be summarized that due to the generally immensely huge complexity of the process of aggregating the complete number of assets and liabilities general solutions (i.e. algorithms) cannot be given. Creativity leading to individual solutions is needed. Using expert knowledge on correlations between the vertices of a A/L risk hierarchy can simplify its structure and less complex forecasts can be achieved. It can lead to more stable results to simulate the aggregated value of two elements than to simulate the value of each elements and aggregate the simulation results. As shall be seen the concept suggested in this paper promotes these ideas by providing the user with the flexibility needed to change the paths of aggregation.

3. Basic Structure of Object-VersPlan

Object-VersPlan offers an object oriented stochastic simulation model of an insurance market and a set of tools for running this model as part of a management game. Within this market several insurance companies which compete with each other follow the instructions of their boards of management with regard to administration, acquisition, reinsurance program, and investment. In order to obtain sufficient understanding for the possibilities of simulating asset/liability profiles according to the previous chapter it is

necessary to have more insight in the principles, components, and tools Object-VersPlan is based upon.³

Principles

Object oriented paradigm

Object Orientation narrows the gap between model and reality: A model following the object oriented paradigm maps reality as a number of objects consisting of *properties* and *methods* (these are the implementations of their abilities) and the *relations* between them. Objects can be described with by their characteristics, capabilities, and behaviour. Characteristics and capabilities are *encapsulated* from being manipulated from outside the object directly. The only way to interact with objects is message-passing. As many objects are of the same kind and share the same capabilities and characteristics, schemes of objects are stored in *classes*. And as some classes are quite similar to each other, classes may be ordered in a class hierarchy like a tree - abstract classes are close to the root of this tree and very specialized classes will be found at the leaves. This *class hierarchy* reduces redundancy as subclasses *inherit* characteristics and capabilities from their superclasses.

The objects in the model are individual *instances* of these object classes. Classes are stored in an object library and serve as component frameworks that stop you from reinventing the wheel through reuse.

It is the relations between objects that create behaviour of the model: "Is_a" relations form the class hierarchy, the information flow between objects generates dynamic communication relations and other relations like "part_of" describe the structure of the model.

Managing complexity

The idea is to manage highly complex systems without paying the price of additional complications. This is put into effect using the following concepts:

- **Model-View-Controller:** When implementing a complex model in software it is helpful to divide the system into three levels:

Models incorporate all the information directly linked to the problem dealt with. They are defined by experts and should approximate reality as closely as possible.

The *View* is the projection of the model: it defines how the user perceives a model. So different information on the same model can be derived from different views on it.

The *Controller* controls interaction between user and system: it receives input and transforms it into commands to the system.

- The system is described by static and dynamic types of submodels: the structure consisting of part_of hierarchies and the processes describing the message flows that steer the simulation activities. As both types of submodels follow the object oriented paradigm it is easily possible to change the number of objects and their relations within each submodel (i.e. its complexity) to the needs of the user.

Meta-Model

The above principles can be put together in a simple and comprehensive meta-model. Here the semantics are defined that are used to describe simulation models in Object-VersPlan.

Components

The class-hierarchy designed for the specific domain of insurance management games is the component framework for constructing custom simulation models. In this case it has been enlarged to incorporate A/L management. This leads to the following subsection of the class hierarchy of model classes as shown below (specific view/controller classes may easily be added on demand).

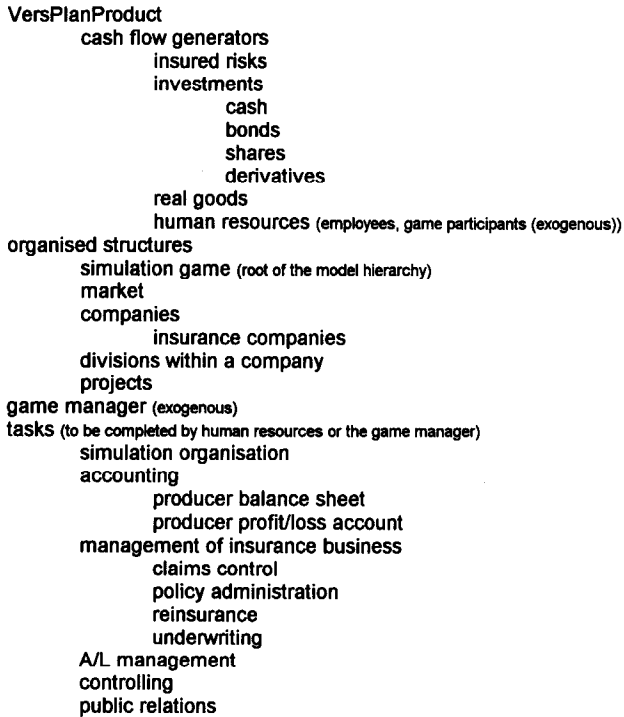


Diagram 3: class hierarchy of model classes

Tools

A rich set of tools supports model construction, simulation handling and implementation of Object-VersPlan for training or management support purposes.

- part-of editor (generates part_of models)
- process editor (generates the simulation processes)
- report editor (interactive generating of user-specific reports)

several simulation support tools (e.g. debugger, notifier, user editor etc.)

The game

Looking at the above possibilities of complexity variation it is obvious that in principle a A/L model can easily be incorporated into Object-VersPlan. Assets and liabilities and also sets of them are objects. Both the composition of portfolios and the simulation instructions can be changed quite naturally as objects and relations between objects can be added, deleted and changed (most of the time the only work to do is to reuse components by "going to the library"). The "views" deliver data in whatever composition and shape the decision taker desires. Thus the object oriented architecture allows for producing realistic learning situations and multiple perspectives on the same facts in a very natural way⁴. These features make it possible to use the same implementation for different training situations. In particular, the effects of "creative" A/L management can be trained and there are options to extend the software to a management support tool. An example of the structure of the whole game can be seen in the following diagram:

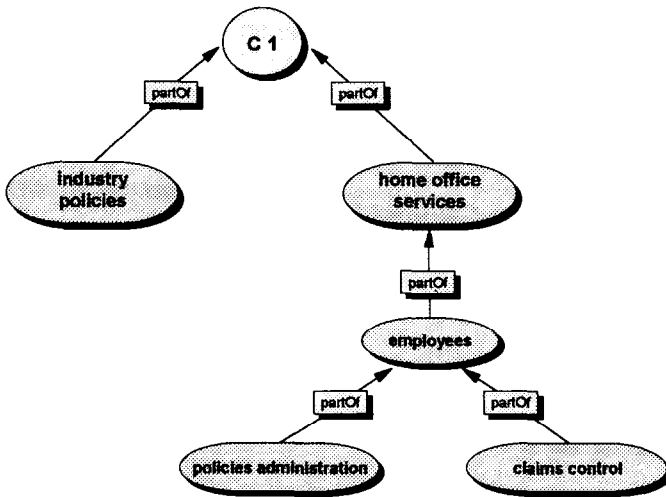


Diagram 4: Example of the structure of an insurance game

An insurance company could look as follows:

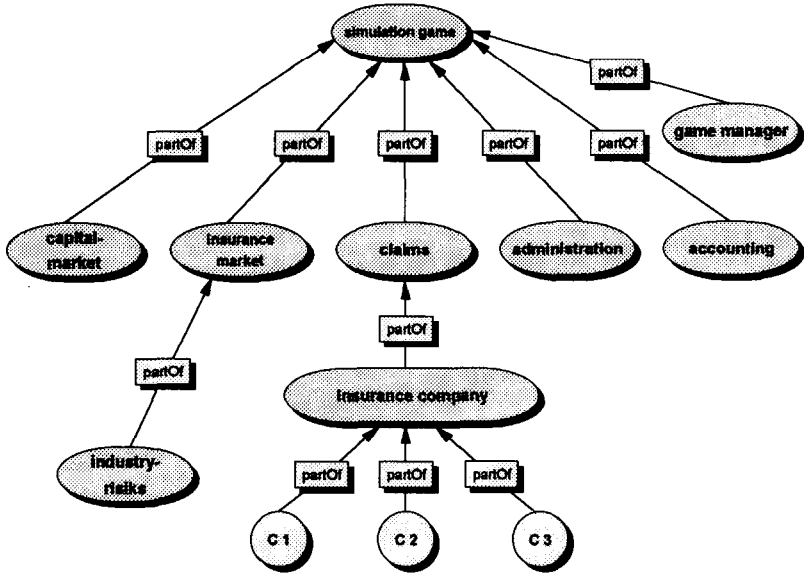


Diagram 5: Example of the structure of an insurance company

Much more can be read into the formulation "an *example* of the structure" than meets the eye. The system has been designed so as to facilitate defining these structures according to the user's every wish (even the above diagrams are not simple drawings but output of the `part_of-editor`). Areas other than insurance could also be modelled (certainly this would mean that the above mentioned libraries would have to be supplemented).

The dynamic character of the model can be seen in an example of the running of one simulation period:

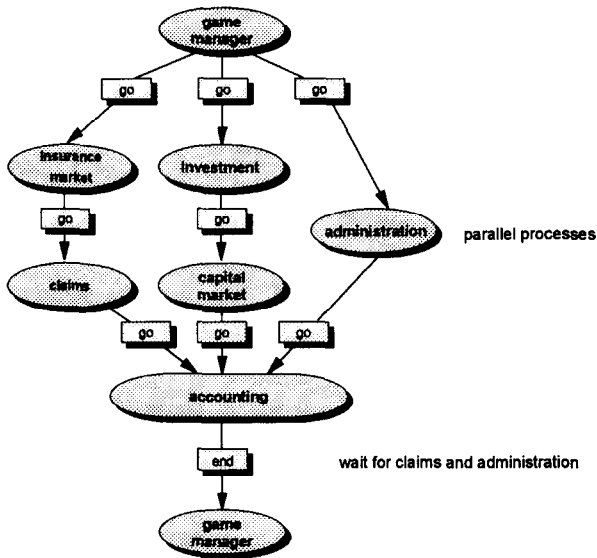


Diagram 6: Example of a simulation period

The diagram shows that during one period several actions can take place simultaneously. Between the periods the "managers" of the companies make their decisions and so define the position of their company within the whole market anew. According to the new state variables the insurance agents take care of the existing business and acquire new risks, others terminate or are lapsed or cancelled. As soon as these developments have been taken care of the claims are simulated and claims are processed. Simultaneously the investments of each company are determined and then the changes in the capital markets are simulated.

4. Example

Object-VersPlan is not simply an abstract tool to model a market. Simulation rules are implemented as well. The user can either make use of them and adapt them to his view on the world or plug in his own models. The following diagrams show a concrete example of the way the future probability distributions of the market value of a portfolio of risks and the corresponding assets can be approximated.

It is assumed the development of the global economy behaves like a deterministic sine function:

$$\Phi(t) = \sin(\omega t)$$

The number of insurance policies a company holds is directly related to this function. Also inflation depends on Φ and determines the average sum assured. Premiums are a constant proportion of sums assured (burning cost), cost a constant proportion of premiums. The number of claims per period follows a Poisson-distribution, claim amount an exponential distribution. The total claim amount per period is derived by Monte-Carlo-simulation. An example of a histogram that approximates the probability distribution of the resulting profit in three years time is given in the next diagram.

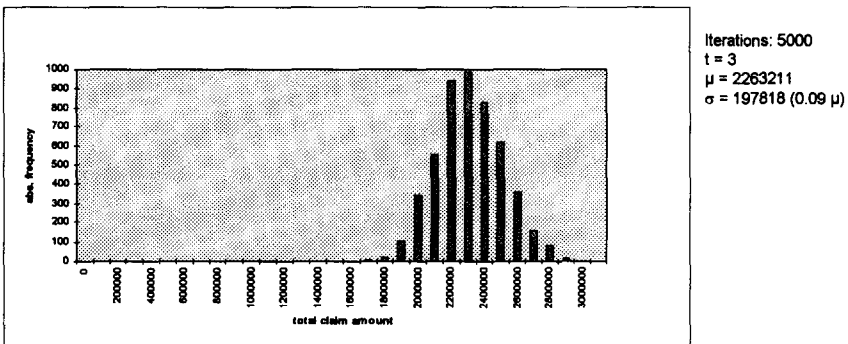


Diagram 7: claims simulation

The available investments are fixed interest bonds with terms to maturity from two to ten years, equities without dividends and options on them. Furthermore money can be put

into an account with a risk free one year interest rate (the risk free asset according to Markowitz). The yields of the bonds with terms to maturity two to nine are deterministic combinations of the one year interest rate and the yield of the ten year bond. These yields follow Φ with mean reversion:

$$r_{\tau}(t) = a_{\tau}(A_{\tau}\Phi(t+\alpha) + c_{\tau}) + (1 - a_{\tau})r_{\tau}(t-1) + \sigma_{r_{\tau}}\varepsilon_{t-1}$$

r : yield to maturity

τ : term to maturity

$a_{\tau}, A_{\tau}, c_{\tau}, \alpha$: mean-reversion parameter

$\varepsilon \sim N(0,1)$

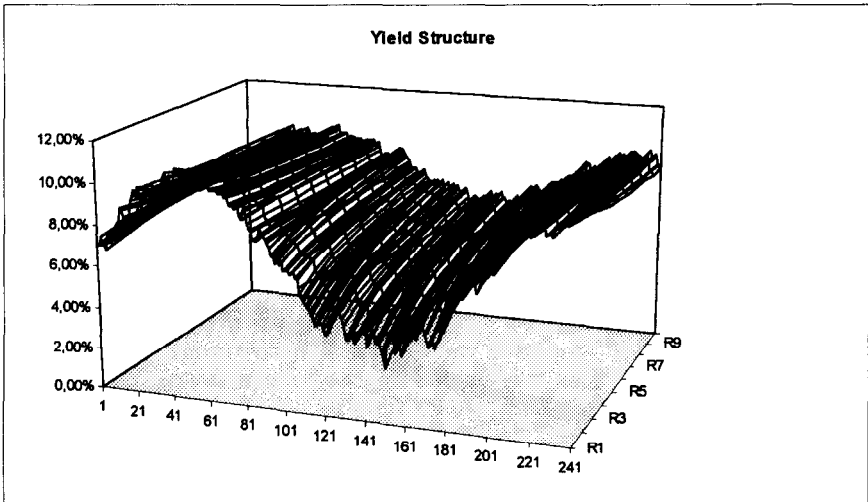


Diagram 8: Example of the bond yield structure for 241 months

The share market has been implemented as a single-factor model according to Sharpe⁵. The single factor is correlated to the yields of the bond market and thus implicitly dependent on Φ ⁶:

$$r_M(t) = r_M(t-1) + g(r_{\tau}(t) - r_{\tau}(t-1)) + \sigma_M\varepsilon_{t-1}$$

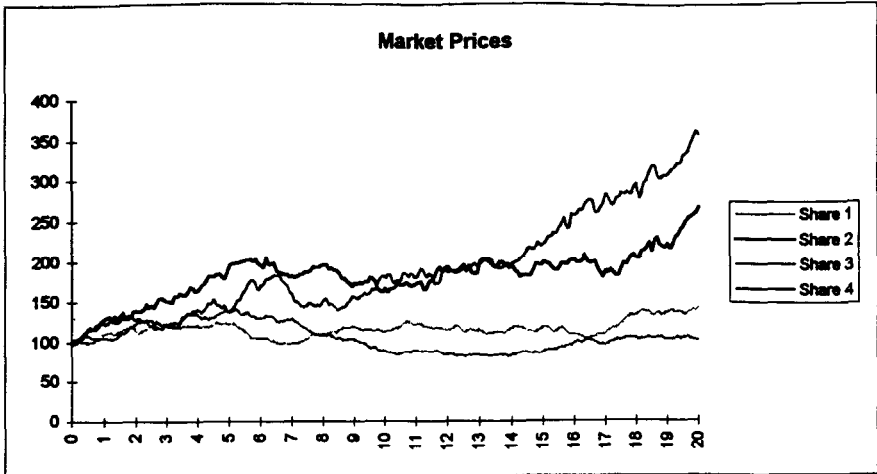


Diagram 9: Example of the development of market prices of shares

It is assumed the market participants agree with the theory that equities follow a random walk and calculate option prices using the Black-Scholes formula.

Under the existing conditions the portfolio managers want to decide between the following (admittedly not very sophisticated) strategies:

I) A constant portion of riskfree asset, bonds and shares is maintained:

shares: 40% (evenly distributed to 4 shares)
 bonds: 45% (evenly distributed to 9 bonds with terms to maturity 2,...,10)
 riskfree asset: 15%

II) Same strategy as I); furthermore:

50% of shares are hedged by short puts
 stop loss with priority 10,000, reinsurance premium burning cost · 1.1

It is assumed the overall market value of all assets in $t=0$ is 1,000,000. The result of the simulation of claims has been shown in diagram 6. The resulting frequency tables are given below.

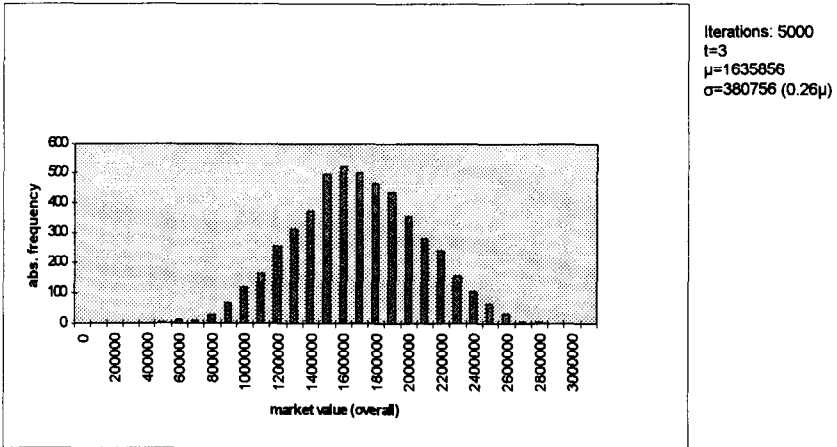


Diagram 9: Results of simulation I

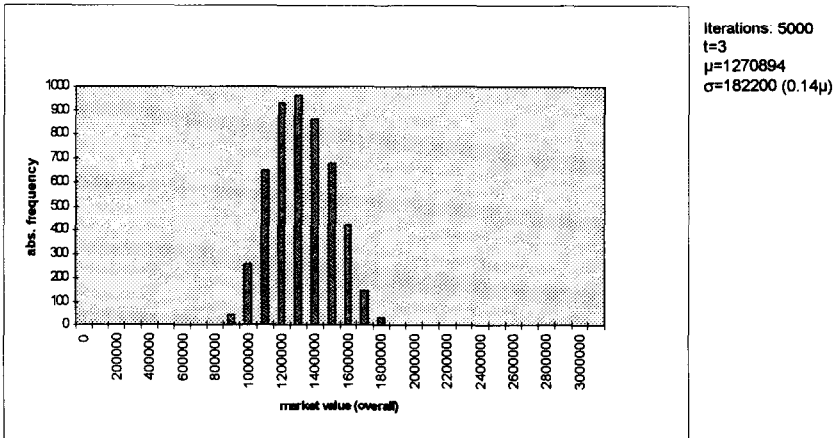


Diagram 10: Results of simulation II

5. Potential And Limitations

If one were bold one could say the potential is unlimited. In theory portfolio managers could be trained at all levels and projections on the effects of all kinds of management decisions are possible. But what are the limitations?

It can be assumed only few exist for Object-VersPlan as a training instrument. Its "classically programmed" predecessor has successfully been employed for some years in several seminars, and the new dimension of complexity variation should lead to at least the same quality of training for other, custom- made situations, levels and goals.

There is no experience so far on its application as a management decision support tool. The quality of the projections it would deliver would be dependent on how well the simulation rules approximate reality. It can be assumed that the currently implemented models would in many cases not yet be able to approximate the user's opinion on the functioning of his insurance portfolios and the capital markets. If he could deliver the missing individual models, these could be substituted for the current model already implemented and so extend the system.

¹ The finding of homogenous sets of assets and liabilities is a complex topic in itself. How this can be done was shown by Elmar Helten (e.g. see Helten, E.: *Die Erfassung und Messung des Risikos*, Wiesbaden, 1994). On an abstract level it can be seen that the same criteria can be applied for both insurance and investment portfolios.

² The terms vertices, edges and leaves refer to those found in graph theory.

³ The following explanations can be found in more detail in "Objektorientierte Planspielentwicklung - dargestellt am Beispiel eines Versicherungsplanspiels" by Ralf Klotzbücher (published in "Versicherung und Risikoforschung", Wiesbaden, 1996) who developed and implemented Object-VersPlan.

⁴ Authentic learning surroundings as a precondition for efficient learning and the benefits of multiple perspectives are the message of constructivism; e.g. see Collins, A., Brown, S., Newman, S.: *Cognitive Apprenticeship*, in: *Knowing, learning and instruction*, New York, 1989, p. 453 ff, and Mandl, H., Gruber, H., Renkl, A.: *Lernen in Schule und Hochschule*, München, 1993, p. 9 ff

⁵ See Sharpe, W.F.: *Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk*, in: *Journal of Finance*, Vol. 19, 1964, p. 425 ff; for helpful comments on its structure and its place in capital markets theory see Alexander, G.J., Sharpe, W.F.: *Fundamentals of Investments*, New Jersey, 1989, and Rudolph, B.: *Kapitalkosten bei unsicheren Erwartungen*, Berlin e.a., 1979

⁶ τ and g allow to determine the correlation between yields of the equity market and the yields of bonds with terms to maturity 1 and 10.