

Актuarный инструментарий¹

1. Введение

Развитие компьютерной техники и программного обеспечения является одним из основных факторов развития страховых продуктов и инвестиционных инструментов. В свою очередь, развитие страховых и инвестиционных продуктов требует соответствующего совершенствования актуарной деятельности. Расчеты, базирующиеся на применении стандартных актуарных функций, все больше вытесняются моделированием денежных потоков, генерируемых портфелем договоров.

Для примера, рассмотрим тарификацию страховых продуктов. В прошлом тарификация традиционных страховых продуктов, проводилась с использованием достаточно простых актуарных формул. Например, для смешанного страхования жизни формула для расчета брутто тарифа могла выглядеть следующим образом:

$$G = \frac{A_{x:\overline{n}|} + a \cdot \ddot{a}_{x:\overline{n}|}}{(1-c) \cdot \ddot{a}_{x:\overline{n}|} - b}.$$

где a , b и c – некоторые заданные параметры нагрузки, или, еще проще,

$$G = \frac{A_{x:\overline{n}|}}{(1-f) \cdot \ddot{a}_{x:\overline{n}|}}.$$

В настоящее время, для установления конкретных значений параметров тарификации, используется основанное на моделировании денежных потоков тестирование прибыльности (profit testing). Более того, многие страховые компании для расчета каждого конкретного значения таблицы страховых тарифов (программа страхования, срок страхования, пол, возраст, порядок уплаты взносов и т.д.) используют тестирования прибыльности.

Расчет резервов для целей государственной отчетности во многих странах, в том числе в России, прогнозирования денежных потоков не требует (см. главу 17). Однако анализ адекватности резервов в соответствии с международной системой финансовой отчетности (МСФО) основан на прогнозировании будущих выплат (в том числе при досрочном расторжении договоров страхования), взносов и расходов.

Любое решение страховой компании, например, выпуск новых страховых продуктов, создание филиалов, принятие инвестиционной стратегии, связано с той или иной неопределенностью будущих финансовых результатов компании, т.е. риском. Для анализа рисков используются модели, позволяющие прогнозировать возможные будущие последствия принятия конкретных решений. На основании предположений о будущем поведении ключевых переменных (т.е. факторов, которые имеют материальное влияние на результаты страховой компании), таких как смертность и/или заболеваемость, доходность инвестиций, издержки, объем нового бизнеса и досрочное расторжение договоров, модели позволяют выбрать

¹ Раздел книги: А. Лельчук Страхование жизни, М. АНКЛ, 450 стр., в печати

вариант дальнейших действий, наилучшим образом соответствующий потребностям компании. В рамках актуарного контрольного цикла моделирование центральным элементом этапа «принятие решения» (см. главу 3).

Разработки сложных актуарных моделей требуют широко распространенная в европейской практике оценка заложенной стоимости страховой компании (embedded value). Еще более сложные задачи будут решаться при подготовке внутренних моделей компании для целей Solvency 2.

Развитие компьютерной техники и программного обеспечения создает базу для совершенствования актуарного инструментария. Вначале, широкое распространение персональных компьютеров и электронных таблиц типа Excel и Lotus123 сделало их основным инструментом актуария. Однако по мере усложнения актуарных задач, электронные таблицы становились все более громоздкими, содержащими большое количество листов, книгами, с трудом поддающимися отладке. Кроме того, многочисленные исследования показывают, что большинство разработанных на базе электронных таблиц приложений имеет, как минимум, одну серьезную ошибку. Проблемы возникают вследствие того, что электронные таблицы очень сложно проверять и интерпретировать.

Дальнейшее развитие программного обеспечения позволило использовать для финансового прогнозирования язык программирования Visual Basic для приложений (VBA), что упростило разработку соответствующих программ и повысило их надежность.

Очередным этапом стала разработка специального инструментария для финансового моделирования, упрощающего подготовку необходимых программ и повышающего их надежность и скорость работы. Первой появилась система Prophet, затем к ней добавились другие широко известные системы, такие как Moses и VIPtech. Эти и иные системы получили заслуженное признание и широко применяются в международной актуарной практике. Вышеуказанные системы предлагают пользователю многотысячные библиотеки функций, из которых актуарий может, как правило, выбрать определенное количество нужных, для решения конкретной задачи и построить необходимую модель. При отсутствии необходимых функций их можно заказать у разработчика системы или написать самостоятельно.

В практике стран с большими традициями страхования жизни применение разработанных на базе Excel (без использования VBA) приложений, предназначенных для решения достаточно сложных актуарных задач, распространено гораздо шире, чем в России. Это связано с так называемой проблемой «наследования»: зачастую бывает трудно отказаться от действующего привычного инструментария и написать новую систему, использующую более современные методы. В российской практике наибольшее распространение получил, если так можно сказать, средний вариант: обычно используются программы, написанные на языке VBA. Сложные большие приложения, подготовленные с использованием спредшитов без применения VBA относительно редки, однако, и современный актуарный инструментарий применяют только дочерние предприятия западных компаний. Поэтому, в данной главе упор будет сделан на сравнительном анализе использования VBA и специализированного актуарного инструментария.

В качестве образца для сравнения используется новый британский актуарный инструмент Mo.net², разработанный на базе Микрософтовской платформы Microsoft.NET³ с использованием VB, реализованного на этой платформе (VB.NET). Mo.net представляет собой:

- 1) современный,
- 2) объектно-ориентированный,
- 3) построенный на платформе .NET,
- 4) использующий VB.NET инструмент.

Эта фраза не случайно разбита на части:

- 1) Современный – учитывающий опыт предшественников, подготовленный с учетом современных актуарных задач, в том числе Solvency 2, свободный от «наследственных» проблем. Под наследственными проблемами понимаются не самые лучшие, устаревшие решения, которые фирма вынуждена продолжать поддерживать.
- 2) Объектно-ориентированное решение обеспечивает гибкость, краткость программного кода, модульность программы: модели и прогнозы являются объектами VB, что обеспечивает простоту их взаимодействия.
- 3) Построенный на платформе .NET обеспечивает высокую скорость расчетов, полную совместимость с приложениями Microsoft, в том числе с Excel и Access.
- 4) Использование VB.NET обеспечивает простоту освоения специалистами, умеющими работать с VB for Excel.

Разработчики Mo.net разработали мощный и гибкий инструмент, который, на взгляд автора, предназначен для актуариев, имеющих некоторый, не обязательно большой, опыт программирования. Они отказались от создания многотысячных библиотек функций в пользу разработки классов объектов, резко упрощающих работу актуария. Для тех, кто знаком с объектно-ориентированным программированием, отметим, что в отличие от VBA, язык VB.NET является полноценным объектно-ориентированным языком программирования. Основными классами объектов Mo.net являются модели, прогнозы и таблицы, с использованием которых мы кратко ознакомимся ниже. Имеется также некоторое количество полезных для разработки приложений функций. Кроме того, можно использовать все функции VB.NET, которые обычно совпадают с функциями VBA.

2. Интерфейс

Многооконный интерфейс Mo.net выглядит так, как будто он является продуктом Microsoft Office, что не удивительно, учитывая, что он создан на базе платформы .NET.

² Произносится на французский лад: Моне, как фамилия знаменитого художника.

³ Все приложения фирмы Микрософт - Windows, Office и т.д. - разработаны на базе платформы .Net. В свою очередь, фирма предоставляет всем желающим возможность использования платформы .Net для разработки своих приложений.

При вызове программы появляется окно приложения (рис.1), а после вызова конкретного проекта, появляется и окно проекта. В окне проекта можно создать новый проект, открыть существующий проект, добавить, удалить объекты, а также скопировать объект из одного проекта в другой.

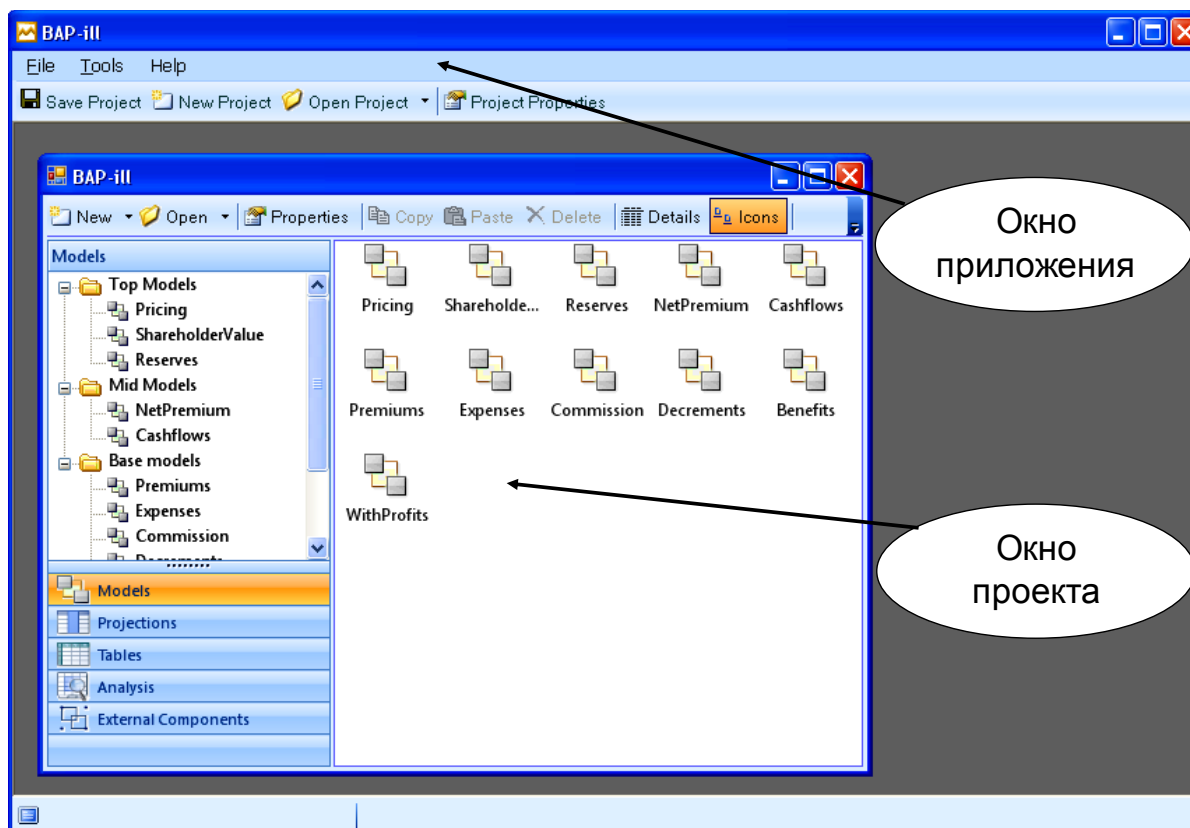


Рис. 1. Окно приложения и окно проекта

На рис. 1 в левом нижнем углу окна проекта нажата кнопка Models, поэтому видны входящие в проект модели. Если будет нажата клавиша Projections, будут видны прогнозы, клавиша Tables - таблицы.

Из окна проекта можно перейти в окно дизайна (Models), окно прогнозов (Projections), окно таблиц (Tables) и окно Анализа результатов (Analysis).

На рис. 2 показано окно таблиц, в котором можно определить и/или изменить свойства таблиц, созданных в окне проекта. В левом верхнем углу показан перечень таблиц, задаваемый в типичной для программ Микрософт древовидной форме. В правой половине окна видна выделенная таблица ProdProfile (для этого нужно щелкнуть на иконке с именем нужной таблицы), а в левом нижнем углу показано окно свойств таблицы, в котором видны свойства выделенной таблицы.

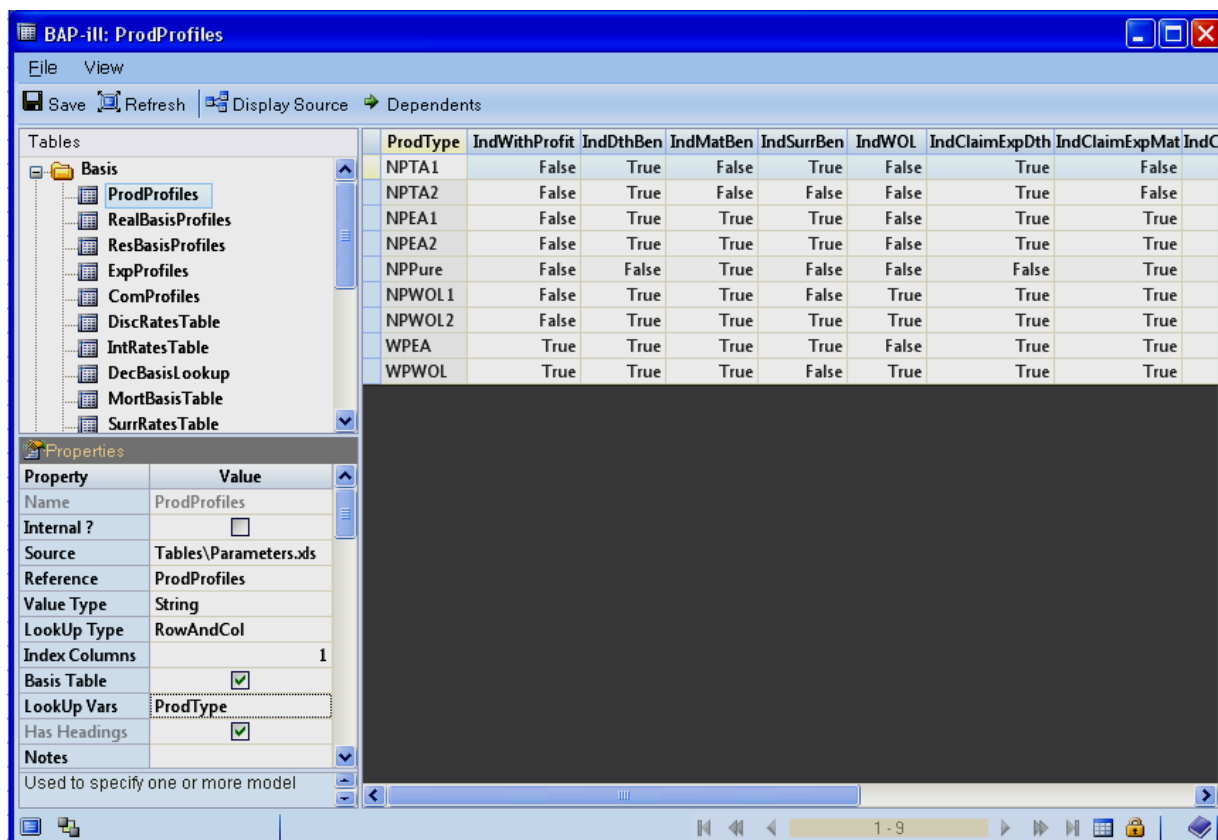


Рис 2. Окно таблиц

Здесь задаются свойства таблицы: источник таблицы, тип, метод поиска по таблице и т.д. Если источником таблицы является текстовая таблица с разделителями, то ее можно отредактировать в данном окне. Таблицы, вызываемые из Excel или Access, редактируются в соответствующих программах.

Окно дизайна (рис. 3) предназначено для создания необходимых для проекта моделей, т.е. задания переменных модели. В Mo.net используются переменные трех типов:

- входные переменные (Inputs), читаемые всеми моделями проекта. В первую очередь они используются для описания полисных данных;
- параметры – переменные, задаваемые для каждой модели;
- функции – функции Visual Basic, которые могут зависеть или не зависеть от времени. Зависящие от времени функции называются столбцовыми, а не зависящие – константными. Типичными примерами столбцовых функций являются резервы, издержки, годовая прибыль, а константных – современная стоимость прибыли и внутренняя норма доходности.

В левом верхнем углу окна дизайна показано дерево моделей, в середине показан, в древовидной форме, список входных переменных, а в правой части – свойства этих переменных в табличной форме. В окне дизайна можно добавить или исключить переменные, скопировать переменные из другой модели, изменить их свойства и т.д. Аналогичным образом в окне дизайна можно работать и с параметрами модели.

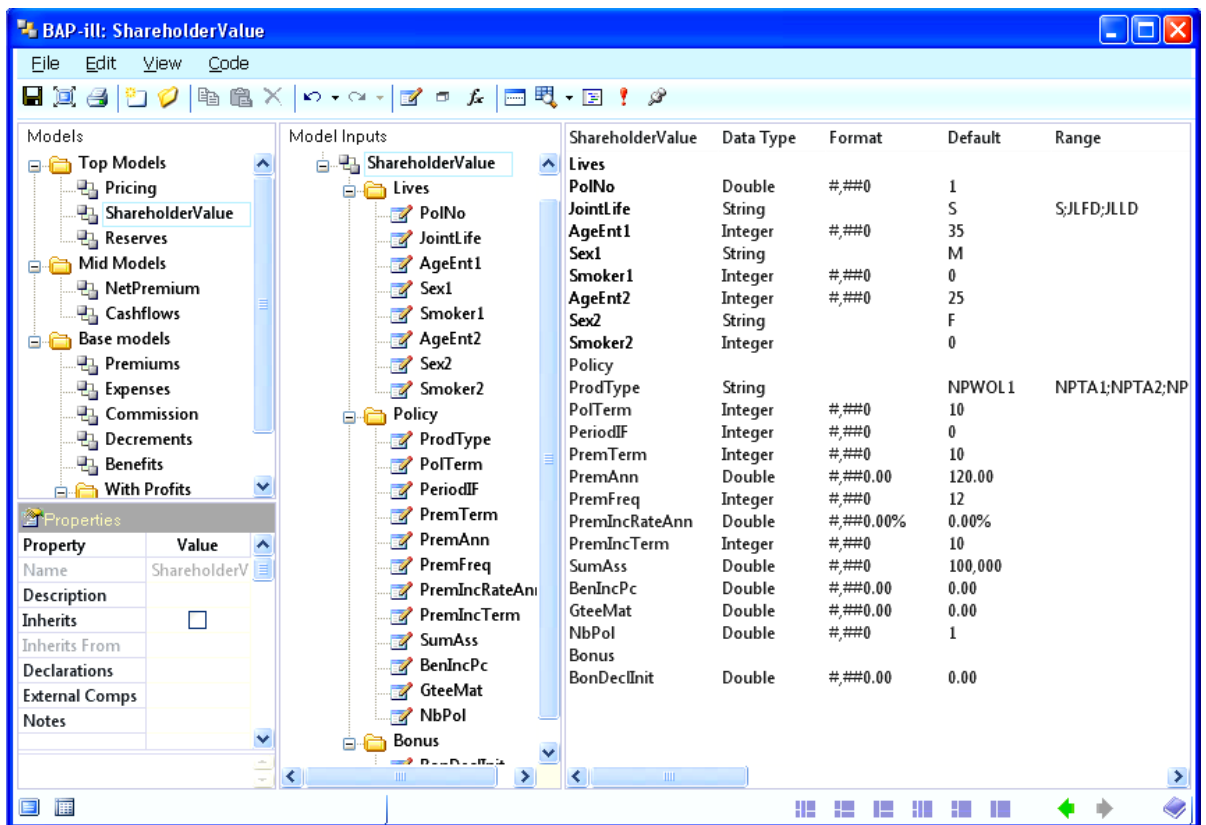


Рис. 3. Окно дизайна, входные переменные

На рис. 4 в окне дизайна показаны функции модели. Аналогично случаю входных переменных в левом верхнем углу видно дерево моделей, в середине – дерево функций. В левом нижнем углу находится окно свойств функции. В данном случае функция имеет имя ResIF, принадлежит к модели ShareholderValue, имеет тип Double и числовой формат с двумя знаками после запятой. В правом нижнем углу расположен редактор формул, предназначенный для написания формул с использованием языка VB.NET.

Когда курсор установлен на имени переменной, в прямоугольнике показываются сведения о данной переменной. На рис. 4 курсор указывает на переменную SubRes, поэтому программа сообщает, что данная переменная является ссылкой на модель Reserves.

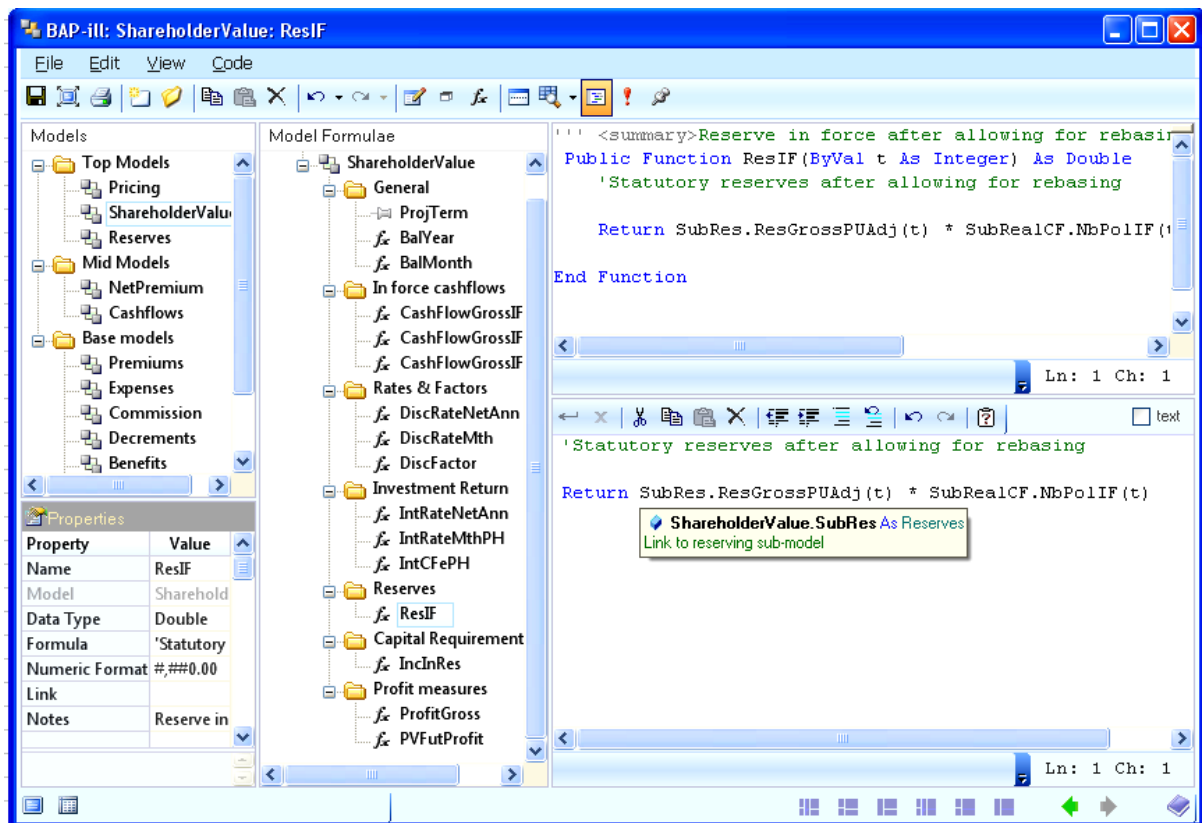


Рис. 4. Окно дизайна, функции

В целях отладки каждая модель может запускаться самостоятельно; для этого входным переменным и параметрам модели обязательно задаются начальные значения. Таким образом, при отладке проводится расчет для одного полиса, задаваемого начальными значениями входных переменных.

Для того чтобы сделать расчет по группе полисов, формируется прогноз, для которого задается таблица полисных данных. На рис. 5 показано окно прогнозов, содержащее (в левом верхнем углу) три прогноза. Справа видна таблица исходных данных выделенного прогноза (ShareholderValue), а в левом нижнем углу – свойства прогноза: имя прогноза, имя модели (в данном случае совпадает с именем прогноза), таблица, содержащая полисные данные и т.д. Отметим, что некоторые графы содержат раскрывающиеся окна. В частности, в раскрывающемся окне Run code можно, например, задать программу, выполняющую данный прогноз несколько раз с разными значениями параметров любой из участвующих в расчете моделей.

Результаты расчета модели или прогноза можно просмотреть в окне анализа (рис. 6). В левой части окна анализа выводятся значения входных переменных, параметров и константных функций. В правой части выводится таблица столбцовых (зависящих от времени) функций. Отметим, что функции выводятся в том же порядке, в каком они были размещены в окне дизайна, причем названия папок используются в качестве подзаголовков таблицы. По умолчанию выводятся суммарные, по всем полисам, значения входных переменных, параметров и функций, что отвечает основной задаче моделирования – получения суммарного денежного потока.

BAP-ill: ShareholderValue

File Run View

Save Print Refresh Compile Run View Output Run Messages

Projections

- Pricing
- ShareholderValue
- Reserves

Properties

Property	Value
Name	ShareholderValue
Copy Properties	(none)
Model	ShareholderValue
Inputs Table	InputData.xls
Variables Set	
Dynamic	<input type="checkbox"/>
Output Settings	IOFiles\Projections\
Build	Debug Info.
Maximum term	1200
Target(s)	
Run code	
Notes	

Inputs	PolNo	NbPol	ProdType	JointLife	AgeEnt1	Sex1	Smoker1	AgeEnt2	Sex2	Smoker2	PoITerm
1	1	1	NPWOL1	S	55	M	1	0		0	0
2	2	1	NPWOL1	S	60	F	1	0		0	0
3	3	1	NPWOL1	S	53	M	0	0		0	0
4	4	1	NPWOL1	S	55	M	1	0		0	0
5	5	1	NPWOL1	S	60	F	1	0		0	0
6	6	1	NPWOL1	S	83	M	0	0		0	0
7	7	1	NPWOL2	S	40	F	1	0		0	0
8	8	1	NPWOL2	S	75	M	0	0		0	0
9	9	1	NPWOL2	S	60	M	1	0		0	0
10	10	1	NPWOL2	S	40	F	0	0		0	0
11	11	1	NPWOL2	JLFD	55	M	1	47	F	0	0
12	12	1	NPWOL2	JLFD	40	F	1	43	M	0	0
13	13	1	WPWOL	S	40	F	0	0		0	0
14	14	1	WPWOL	S	34	M	0	0		0	0
15	15	1	WPWOL	S	22	F	0	0		0	0
16	16	1	WPWOL	S	40	F	1	0		0	0
17	17	1	WPWOL	S	34	M	1	0		0	0
18	18	1	WPWOL	S	48	F	1	0		0	0
19	19	1	WPWOL	S	22	M	0	0		0	0
20	20	1	WPWOL	S	33	F	0	0		0	0
21	22	1	WPWOL	JLFD	33	F	0	31	M	0	0
22	23	1	WPWOL	JLFD	48	F	1	56	M	1	0
23	23	1	WPWOL	JLFD	50	M	0	36	F	1	0
24	24	1	WPWOL	JLFD	34	M	1	32	F	0	0

Inputs Constants Input and Constants

Sum=245

Рис. 5. Окно прогноза

BAP-ill: ShareholderValue

File View Code Reporting

Save Treeview Precedents Dependents Compare Compile Run Graph Export

Inputs	Value	Periods		Investment Return		Reserves	Capital Requirement	Profit measures	
		Month	Year	IntRateMthPH	IntCFEPh	ResIF	InclnRes	ProfitGross	PVFutProfit
PolNo	302	0	0	7.66%	0.00	0.00	0.00	0.00	16.87
JointLife		1	0	7.47%	-10.30	13,840.52	13,840.52	-17,243.22	17,259.46
AgeEnt1	1,114	2	0	7.47%	44.40	13,695.83	-144.70	482.05	16,832.47
Sex1		3	0	7.47%	43.95	13,550.60	-145.23	477.90	16,408.25
Smoker1	13	4	0	7.47%	44.38	13,404.83	-145.77	759.56	15,700.98
AgeEnt2	245	5	0	7.47%	43.03	13,258.51	-146.32	468.67	15,282.16
Sex2		6	0	7.47%	42.57	13,111.63	-146.88	466.12	14,864.53
Smoker2	2	7	0	7.47%	49.10	13,931.76	820.12	1,749.63	13,162.02
Params	Run1	8	0	7.47%	44.66	13,776.96	-154.80	465.48	12,738.12
ProdProfiles	ProdProfiles	9	0	7.47%	44.17	13,621.35	-155.61	463.24	12,315.08
ProdDiscGrp	DiscGrp2	10	0	7.47%	44.54	13,674.90	53.55	528.51	11,825.38
RealBasProfile	Basis1	11	0	7.47%	43.84	13,517.42	-157.48	458.70	11,403.90
RealBasisProfiles	RealBasisProfiles	12	1	7.47%	43.35	13,359.12	-158.30	456.51	10,983.22
ResBasProfile	Basis1	13	1	7.28%	66.11	18,540.86	5,181.74	3,159.37	7,858.30
DecProfile	Dec1	14	1	7.28%	57.43	18,349.33	-191.53	450.63	7,431.48
ExpProfile	Exp1	15	1	7.28%	56.84	18,157.29	-192.03	448.40	7,005.51
ComProfile	Com1	16	1	7.28%	57.08	18,174.57	17.24	560.02	6,516.76
DiscProfile	Disc1								
DiscRatesTable	DiscRatesTable								
IntBasis	Int3								
IntRatesTable	IntRatesTable								
ValDate	31/12/2007								
IndBonFnl	<input checked="" type="checkbox"/>								
SubRes	Reserves								
Constants	Run1								
ProjTerm	20,796								

'Gross shareholder profit

```

If t > 0 Then Return CashFlowGrossIFCFb(t) + CashFlowGrossIFCFm(t) _
+ CashFlowGrossIFCFe(t) + IntCFEPh(t) - IncInRes(t)

```

ShareholderValue: ProfitGross: -17243.216266026 Ln: 1 Ch: 1

ShareholderValue SubRes SubRealCF

-17243.216266026

Рис. 6. Окно анализа

Если нажать кнопку дерево результатов (выделена на рис. 7 стрелкой), слева можно увидеть структуру модели. Щелчок на названии любой показанной в дереве результатов модели позволяет увидеть результаты работы этой модели. В частности, на рис. 7 видны результаты работы модели Expenses (издержки); в самом верху окна показана цепочка вызовов данной модели.

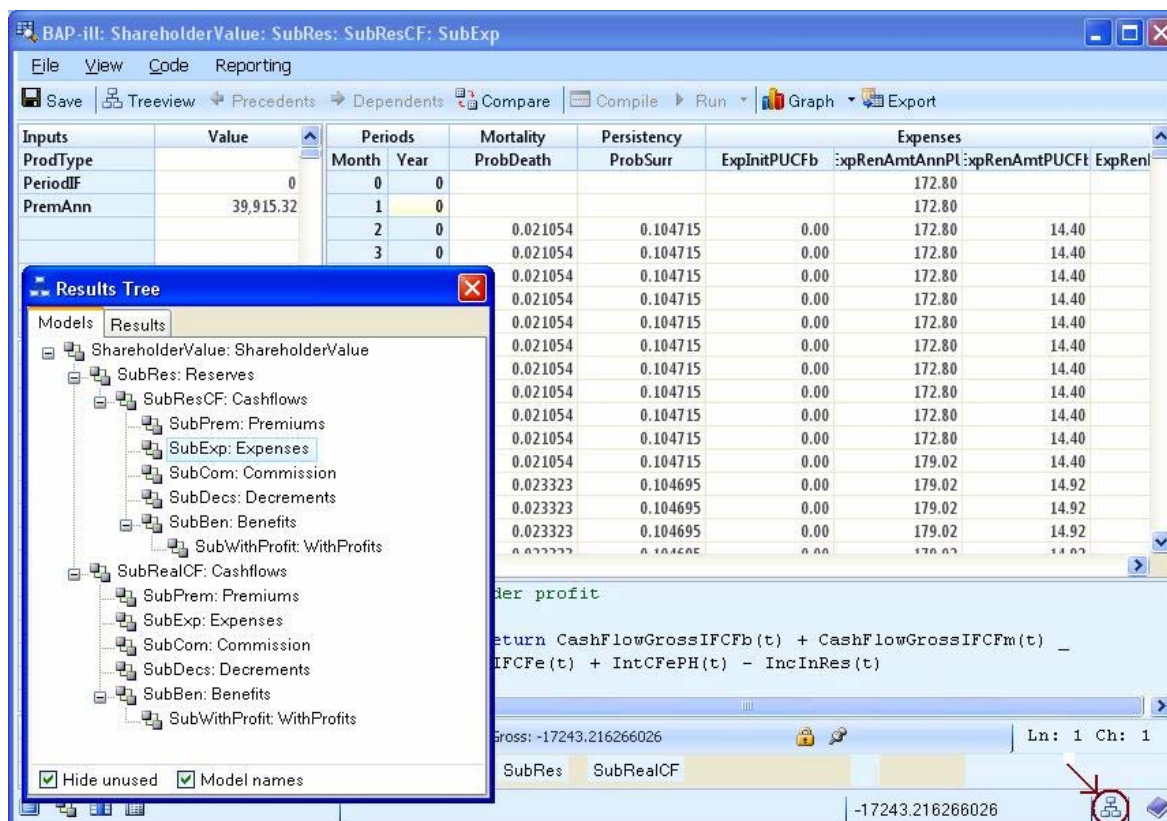


Рис. 7. Окно анализа с деревом результатов (слева)

3. Таблицы

Mo.net предоставляет пользователю простые и удобные средства работы с данными, сохраненными в табличной форме. Данные могут храниться в текстовом файле с разделителями, в Excel или Access. Простоту чтения табличных данных проиллюстрируем на базе данных, хранящихся в Excel.

Первым этапом является создание объекта таблица в окне проекта. Для этого, нужно просто указать его имя. Свойства объекта задаются в окне таблиц (рис. 2). В окне свойств (левый нижний угол экрана) в графе источник (Source) указывается книга Excel, в которой хранится таблица. В данном случае это книга Parameters.xls, расположенная в директории Tables. В графе ссылка (Reference) указывается именованная область Excel, в которой расположена таблица, ProdProfiles. В графе Value Type указывается тип хранящихся данных, в данном случае числовой, Double. В графе LookUp Type указывается тип поиска по таблице, в данном случае поиск ведется по строкам и столбцам (RawAndCol), а в графе Index Columns – количество индексных столбцов, т.е. столбцов, по которым будет производиться поиск. В данном случае поиск будет производиться по одному индексному столбцу ProdType, который определяет строку поиска. Столбец поиска будет определять имя столбца или его номер по порядку. Например, если ProdType=NPTA1, то запрос

ProdProfile.Value(ProdType, "IndDthBen")

вернет значение "True", находящееся на пересечении первой строки, определяемой значением переменной ProdType и второго столбца, имя которого указано в запросе. Объект таблица имеет также ряд полезных свойств, например, количество строк и столбцов в таблице.

У таблицы ProdProfiles стоит галочка в графе Basis Table. Это означает, что данная таблица объявлена в качестве базовой. Базовые таблицы используются для автоматического задания значений параметров модели. Заголовками столбцов данной таблицы являются имена параметров модели. Для того чтобы присвоить некоторому параметру модели значение, зависящее от других переменных, достаточно указать в его свойствах ссылку на соответствующую базовую таблицу. На рис. 8 в левом нижнем углу показаны свойства параметра IndWithProfit. В качестве источника (Source) указана таблица ProdProfiles. Это означает, что значение переменной IndWithProfit будет определяться из столбца с одноименным заголовком в соответствии со значением переменной ProdType.

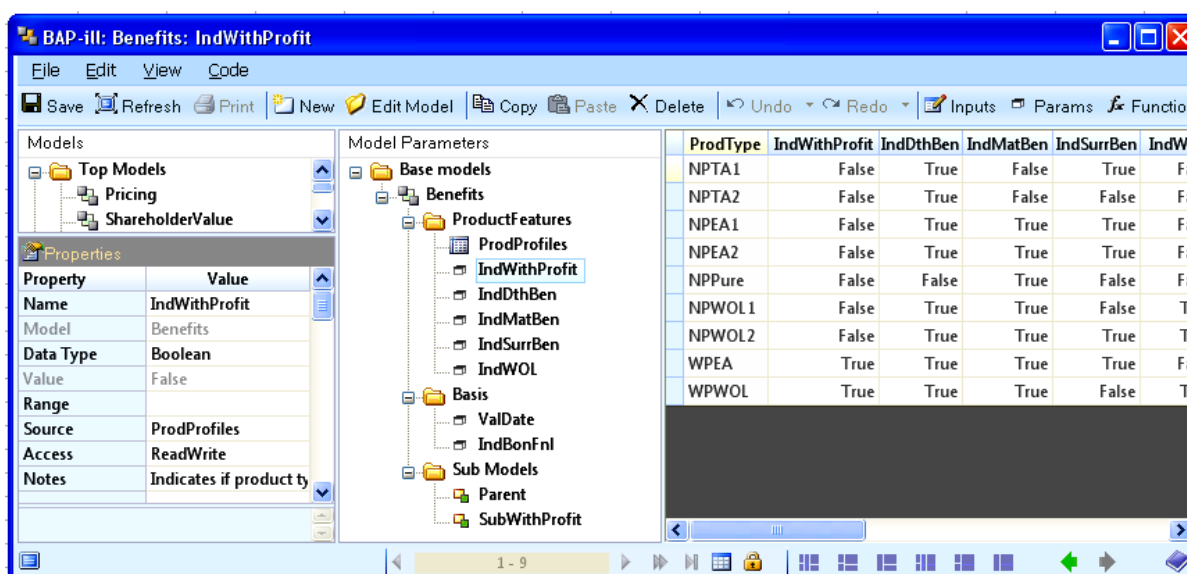


Рис. 8. Окно дизайна. Определение параметра ссылкой на базовую таблицу

Переменная ProdType обозначает имя страхового продукта, например, NPTA1 (Non Profit Term Assurance 1) означает не участвующее в прибыли страхование на срок, вариант 1. Таблица ProdProfile содержит описание (профиль) продукта. Булевские (правда, ложь) переменные определяется участие в прибыли (IndWithProfit), наличие пособия по смерти (IndDthBen) и по дожитию (IndMatBen), и другие параметры. Проект содержит еще несколько базовых таблиц:

- RealBasisProfiles описывает реалистичный базис моделирования,
- ResBasisProfiles описывает базис предписанного резервирования
- ExpProfiles описывает издержки (см. табл. 1)
- ComProfiles описывает комиссионное вознаграждение.

Отметим, что таблица ExpProfiles содержит два индексных столбца – ExpProfile и ProExpGrp, значения которых, в свою очередь определяются базовой таблицей ProdProfiles.

Таблица ExpProfile, описывающая издержки страховой компании

ExpProfile	ProdExpGrp	ExplnitAmt	ExplnitPc	ExpRenAmtAnn	ExpRenPc	ExpClaimAmt	ExpInfIPc
Exp0	ExpGrp1	0.00	0.0%	0.00	0.00%	0.00	0.00%
Exp0	ExpGrp2	0.00	0.0%	0.00	0.00%	0.00	0.00%
Exp0	ExpGrp3	0.00	0.0%	0.00	0.00%	0.00	0.00%
Exp1	ExpGrp1	10.00	15.0%	2.40	1.00%	30.00	3.00%
Exp1	ExpGrp2	50.00	20.0%	6.00	1.50%	40.00	2.00%
Exp1	ExpGrp3	100.00	25.0%	12.00	2.00%	50.00	3.00%
Exp2	ExpGrp1	12.00	18.0%	2.88	1.20%	36.00	3.60%
Exp2	ExpGrp2	60.00	24.0%	7.20	1.80%	48.00	2.40%
Exp2	ExpGrp3	120.00	30.0%	14.40	2.40%	60.00	3.60%

4. Модели

Модели состоят из входных переменных, параметров и функций, определенных пользователем. Они также содержат функциональные возможности в виде предоставляемых системой методов и системных переменных. Новые модели создаются в окне проекта, а их «наполнение» - входные переменные, параметры и функции создаются в окне дизайна.

На рис. 3, 4 и 8 показан вид окна дизайна. Переменные, параметры и функции проекта сгруппированы в удобном для разработки и отладки (чтения) программы виде. На рис. 4 мы видим расчет столбцовой (зависящей от времени t) функции. Отметим, что цикл по времени t задается в неявном виде указанием, что функция зависит от системной переменной $t - f(t)$. Если функция не содержит параметра t , то она является константной, то есть не зависящей от времени. Параметр t может быть определен в месяцах, кварталах, полугодиях или годах.

Зависящая от времени функция может рассчитываться «вперед»: от выпуска полиса до конца срока его действия или конца периода прогноза. Для рекурсивного расчета современной стоимости будущих денежных потоков может потребоваться расчет «назад», скажем от конца срока действия договора до его выпуска. Система сама определяет направление расчета на основании кода функции. Приведенная ниже функция рассчитывает современную стоимость будущих выплат по смерти в предположении, что выплата производится в конце каждого полисного года.

If $t = \text{ProjTerm}()$ Then Return 0 (1)

$$\text{PVDthBen}(t) = (\text{PVDthBen}(t + 1) * (1 - \text{Qx}(t)) + \text{Qx}(t)) / (1 + \text{IntTech})$$

Поскольку современная стоимость выплат $\text{PVDthBen}(t)$ на момент времени t определяется в «обратном направлении» на основании современной стоимости выплат на момент времени $t+1$, система автоматически будет вести расчет «назад». Отметим, что рекурсивные функции такого типа позволяют за один прогон вычислить необходимые для расчета резервов современные стоимости будущих выплат для всех значений t , что многократно сокращает объем счета по сравнению с использованием актуарных функций типа:

$$A_{x+t:n-t}^1$$

расчет значения которой для каждого значения t требует времени, сопоставимого с временем работы рекурсивной формулы.

Таким образом, отказ от написания огромных библиотек функций обусловлен двумя факторами:

- простотой написания нужной функции в VB;
- более высокой эффективностью рекурсивных функций, по сравнению со стандартными актуарными функциями.

Система поддерживает большое количество типов параметров. В частности, параметр может иметь тип *модель* или *проект*, что обеспечивает очень простые методы доступа к параметрам и результатам расчета других моделей и даже других прогнозов. На рис. 4 показан вызов значений функции из другой модели: выражение `SubRes.ResGrossPUAdj(t)` вызывает зависящее от времени t значение функции `ResGrossPUAdj(t)` модели `Reserves`; параметр `SubRes` имеет тип *модель* и значение `Reserves`. Столь же просто было бы вызвать значение функции или параметра другого прогноза.

Функции модели расположены в порядке, удобном для чтения программного кода. А как же быть с порядком расчета? Порядок расчета функций определяется системой самостоятельно, на основании кода программы. Вызов других моделей происходит в случае запроса значения ее параметра или функции; в рассмотренном выше примере имел место вызов модели расчета резервов `Reserves`. Более того, в вызываемой модели будут проведены только необходимые для ответа на все существующие в программе вызовы данных из этой модели; ненужные расчеты не проводятся.

Система предоставляет ряд удобных средств отладки и анализа моделей. Например, можно посмотреть дерево переменных, от которых зависит та или иная функция, а также дерево функций, которые зависят от той или иной переменной. Для этого достаточно нажать кнопку `Precedents` (предшествующие) и `Dependents` (зависящие) – см. рис. 9 и 10. Средства отладки программы в `Mo.net` более мощные, чем в `Excel`.

Полезным и мощным средством отладки является сравнение моделей, для запуска которого достаточно указать имена сравниваемых моделей. В разрезе входных переменных, параметров и функций система показывает три группы отличий:

- переменные, которые есть в обеих моделях, но определены по-разному;
- дополнительные переменные, которых нет в сравниваемой модели;
- отсутствующие переменные, т.е. переменные, которых нет в «основной» модели.

Пример сравнения двух близких по составу моделей показан на рис. 11. В отношении параметров можно сказать следующее:

- по-разному определен ссылающийся на таблицу параметр `RowAndCol`: различаются значения источника (`Source`) и доступа (`Access`);
- параметр `Loading` имеет разные значения;
- «основная» модель имеет один дополнительный параметр `ZilmerFactorCV` и ни одного «недостающего».

Сравнение функций показало:

- различия в двух функциях V и NP;
- наличие трех дополнительных функций.

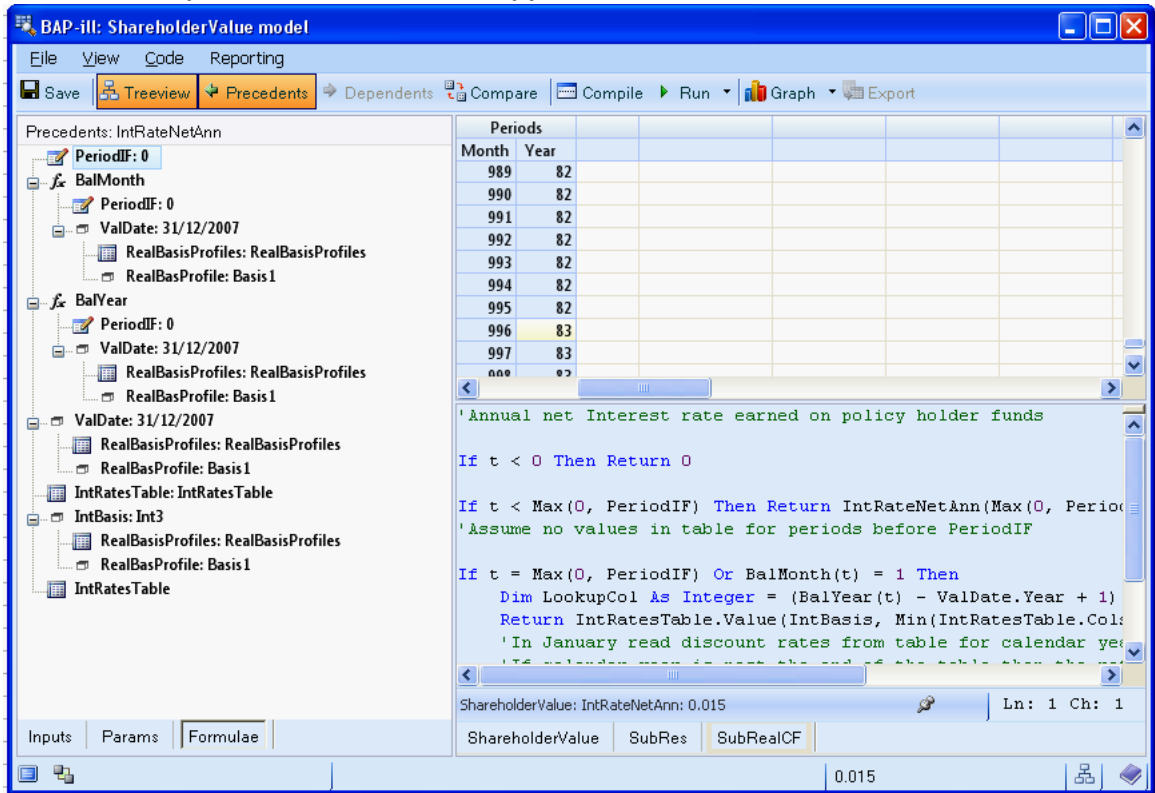


Рис. 9. Предшествующие переменные

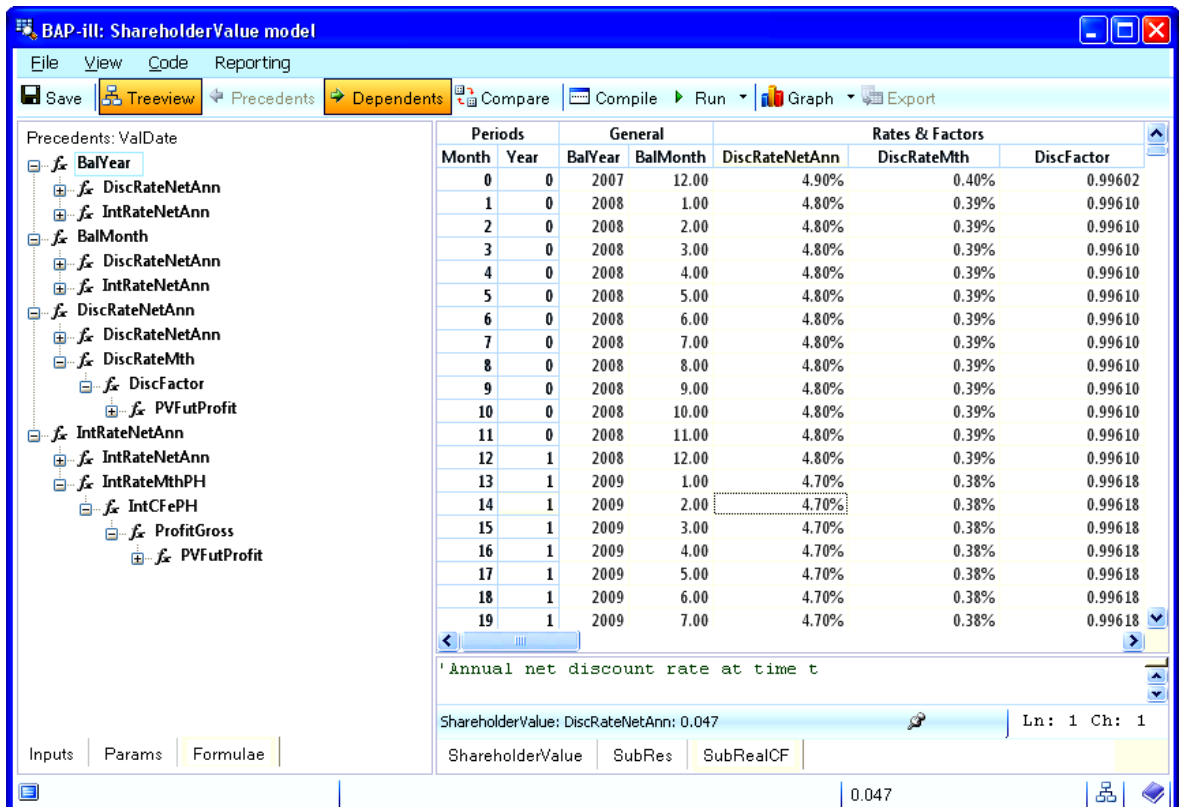


Рис. 10. Зависимые переменные

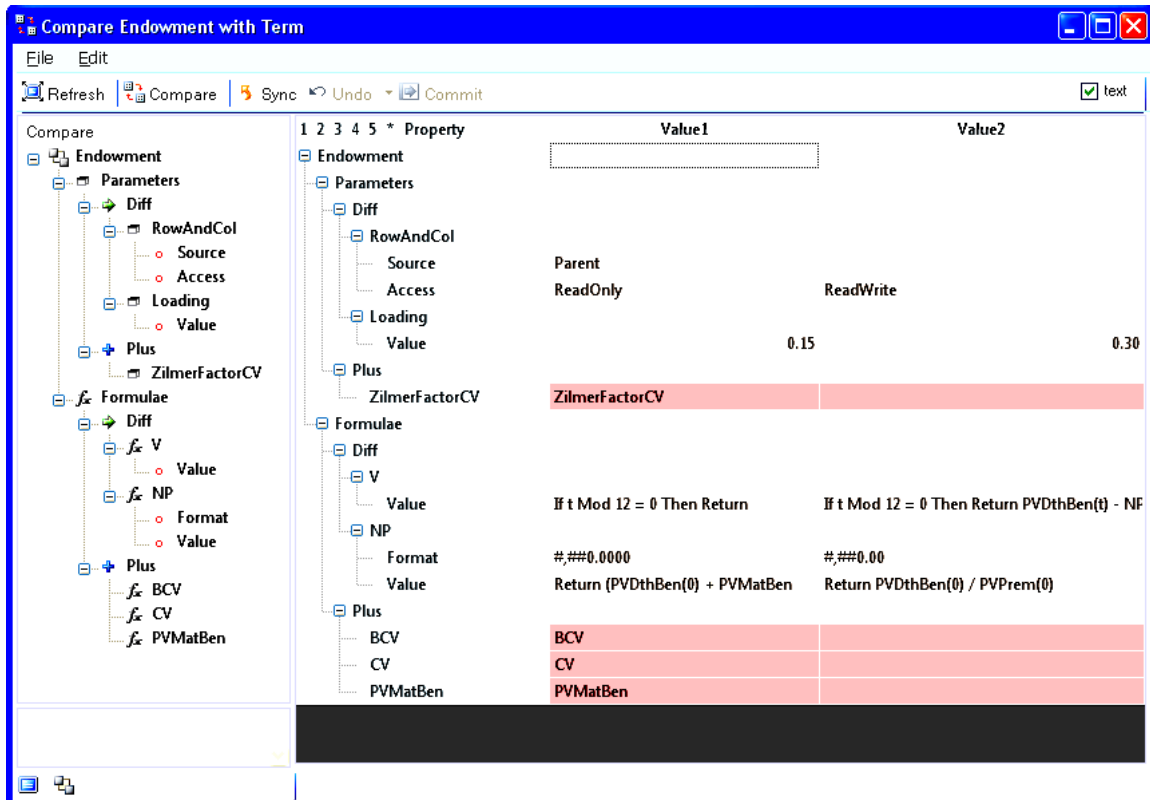


Рис. 11. Сравнение моделей

5. Прогнозы

Система предусматривает несколько типов прогнозов:

- Индивидуальный прогноз
- Преобразование данных
- Групповой прогноз
- Динамический прогноз

Индивидуальный или стандартный прогноз используется для выполнения расчетов, задаваемых некоторой моделью. Прогноз использует эту модель вместе с таблицей входных переменных, которая, как правило, представляет собой полисные данные. Индивидуальный прогноз проводит определенные моделью расчеты для каждой строки таблицы входных данных и выводит суммарные, по всем строкам, значения функций и входных переменных. Прогноз также позволяет производить некоторые дополнительные операции, например, фильтрацию данных, запрос результатов по подгруппам, спецификацию вывода результатов, организацию параллельной работы на нескольких компьютерах и т.д.

Преобразование данных применяется для приведения формата данных в форму, пригодную для индивидуального прогноза. Например, дата рождения может быть преобразована в возраст, буквенные обозначения пола заменены на числовые. Преобразование применяется для группировки данных, позволяющей, в частности, создать таблицу, основанную на модельных точках. Количество записей можно, также сократить за счет слияния «наименее значимых» записей с другими записями.

Таблица 2. Пример полисных данных

Пол	Возраст	Премия	Частота уплаты в год
M	36	30	12
M	42	240	1
F	56	50	12

Преобразование пола в числовой формат за счет замены "M" и "F" на 0 и 1, и преобразование премии к годовому размеру, приведет к таблице.

Таблица 3. Преобразованные полисные данные

Пол	Возраст	Годовая премия
0	36	360
0	42	240
1	56	600

Группировка по полу с вычислением средних значений для возраста и Годовой премии и добавлением количества полисов создает таблицу:

Таблица 4. Группированные полисные данные

Пол	Возраст	Годовая премия	Количество
0	39	300	2
1	56	600	1

Групповой прогноз может использоваться для выполнения любого корректного кода Vb.NET. Обычно групповой прогноз оперирует одним или группой индивидуальных прогнозов, обращаясь с ними как с программируемыми объектами. Например, приведенная ниже подпрограмма проводит вычисления для прогноза Term при пяти значениях параметра ValRate.

```
Sub Run()
    For ValRate As Double = 0.01 to 0.05 Step 0.01
        Term.ValRate = ValRate
        Term.Run()
    Next
End Sub
```

Динамический прогноз используется в случаях, когда нужно периодически принимать решения, основанные на результатах прогноза, например, если инвестиционная политика компании предусматривает изменение распределения инвестируемых активов в зависимости от соотношения активов и обязательств.

Mo.net предусматривает разные варианты динамического моделирования активов и обязательств. Ниже приведен пример, использующий групповой прогноз. Проект содержит два прогноза. Прогноз Assets вычисляет размер активов, прогноз Liabs – обязательств (резервы). Assets содержит параметр EqProp, определяющий долю активов, инвестируемых в акции (остаток

инвестируется в облигации), и функцию `AssetValue`, вычисляющую общий размер активов в каждый момент времени прогноза. Прогноз `Liabs` содержит функцию `LiabValue`, вычисляющую обязательства. Для реализации динамического моделирования, использующего эти два прогноза достаточно следующего кода:

```
Sub Run()  
    For Time As Integer = 0 To 60  
        Asset.EqProp = EqProp(Time)  
    Next  
    Liab.WriteResults: Asset.WriteResults  
End Sub  
  
Function EqProp(Time As Integer) As Double `Доля инвестиций в акции  
    If Liabs.Column("LiabValue", Time) > 1.10 * Assets.Column("AssetValue", Time) Then  
        Return 0.60  
    Else  
        Return 0.40  
    End If  
End Function
```

Подробное объяснение работы данного кода выходит за рамки данной книги. Отметим лишь, что при вычислении функции `EqProp(Time)` запрашиваются значения обязательств и активов, вычисляемые прогнозами `Liabs` и `Assets`. Этого запроса достаточно для запуска соответствующих прогнозов.

6. Результаты

Результаты расчетов хранятся на диске в текстовых файлах с разделителями и их можно просмотреть в окне анализа (рис. 6 и 7). С точки зрения объема программирования принципиально важно то, что таблицы результатов формируются автоматически. При работе в VBA программирование вывода результатов расчетов всех моделей является весьма трудоемкой задачей.

В окне анализа для каждой расчетной величины можно посмотреть код вычисляющей ее функции. Более того, при отладке программы в окне анализа можно изменить не только значения параметров или входных переменных, но и код функций и повторно запустить их на выполнение.

Одним нажатием кнопки всю таблицу можно скопировать на лист Excel. Копирование в Excel производится посредством набора функций VBA, которые `Mo.net` пересылает в модуль VBA (см. рис. 12).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Workspace	D:\Documents and Settings\LAL-R\LALHOME-R\MONET\BAP-ill																		
2	Projection	ShareholderValue																		
3	Results	D:\Documents and Settings\LAL-R\LALHOME-R\MONET\BAP-ill\Files\Projections\																		
4																				
5	Inputs				BalYear	BalMonth	IntCFePH	ResIF	InclnRes	ProfitGross	PVFutProfit									
6	PolNo	302		0	2007	12.00	0.00	0.00	0.00	0.00	16.87									
7	JointLife	0		1	2008	1.00	-10.30	13,840.52	13,840.52	-17,243.22	17,259.46									
8	AgeEnt1		1,114	2	2008	2.00	44.40	13,695.83	-144.70	482.05	16,832.47									
9	Sex1	0		3	2008	3.00	43.95	13,550.60	-145.23	477.90	16,408.25									
10	Smoker1		13	4	2008	4.00	44.38	13,404.83	-145.77	759.56	15,700.98									
11	AgeEnt2		245	5	2008	5.00	43.03	13,258.51	-146.32	468.67	15,282.16									
12	Sex2	0		6	2008	6.00	42.57	13,111.63	-146.88	466.12	14,864.53									
13	Smoker2		2	7	2008	7.00	49.10	13,931.76	820.12	1,749.63	13,162.02									
14	ProdType	0		8	2008	8.00	44.66	13,776.96	-154.80	465.48	12,738.12									
15	PolTerm		0	9	2008	9.00	44.17	13,621.35	-155.61	463.24	12,315.08									
16	PeriodIF		0	10	2008	10.00	44.54	13,674.90	53.55	528.51	11,825.38									
17	PremTerm		389	11	2008	11.00	43.84	13,517.42	-157.48	458.70	11,403.90									
18	PremAnn		39,915.32	12	2008	12.00	43.35	13,359.12	-158.30	456.51	10,983.22									
19	PremFreq		87	13	2009	1.00	66.11	18,540.86	5,181.74	3,159.37	7,858.30									
20	PremIncRateAnn		37.00%	14	2009	2.00	57.43	18,349.33	-191.53	450.63	7,431.48									
21	PremIncTerm		96	15	2009	3.00	56.84	18,157.29	-192.03	448.40	7,005.51									
22	SumAss		677,000	16	2009	4.00	57.08	18,174.53	17.24	509.82	6,516.76									
23	BenIncPc		0.22	17	2009	5.00	56.30	17,981.24	-193.29	443.87	6,092.38									
24	GteeMat		150,000.00	18	2009	6.00	55.71	17,787.42	-193.82	441.69	5,668.83									
25	NkPol		21	19	2009	7.00	61.79	18,694.56	907.14	1,543.28	4,142.32									

Рис. 12. Результаты расчетов в Excel

Из рис. 12 видно, что в ячейке S12 находится не число, а значение функции ColumnVal(Results; Projection; S\$5; D10). Функция ColumnVal предназначена для чтения значений столбцовых функций. Первый параметр Results ссылается на именованную клетку B3, содержащую адрес, где хранятся результаты расчета; второй параметр Projection задает имя прогноза (клетка B2); третий параметр S\$5 задает имя столбцовой функции (PVFutProfit), четвертый \$D10 – читает значение переменной времени. Аналогичными функциями на лист выводятся значения входных переменных, параметров и независящих от времени функций.

Нетрудно догадаться, что с использованием этих функций в Excel нетрудно создать выходной отчет любой формы.

7. Запуск Mo.net из Excel

Когда необходимые программы полностью отлажены, и в Excel сформированы простые и информативные формы отчетности, дальнейшую работу можно проводить не выходя из Excel. Запуск прогноза Mo.net из VB for Excel очень прост; напомним, что совместимость этих двух программных продуктов обеспечивает Микрософт. В простейшем случае достаточно запуска следующей программы:

```

Sub Run
  Dim Project As New GenExcelReporting.Project
  Project.Workspace = Range("WorkSpace").Text
  With Project.Projection(Range("Projection").Text)
    .Run
  End With

```

End Sub

Не выходя из Excel можно изменять ссылку на таблицу входных переменных (полисных данных), менять значения параметров и т.д. Разумеется, для этого, программу запуска придется несколько расширить, однако написание такого рода программы для человека, знакомого с VBA никаких проблем не составит.

Заключение

Подытожим некоторые достоинства работы со специализированным актуарным инструментарием по сравнению с использованием VB для Excel:

- многократно более высокая скорость счета:
 - компилированный VB.NET работает со скоростью, во много раз превосходящей скорость VBA;
 - Запись данных в текстовые файлы несопоставимо быстрее записи в таблицы Excel;
- возможность продолжения работы с системой во время счета, не снижая производительности системы;
- возможность параллельной работы на нескольких компьютерах;
- высокая надежность результатов и простота аудита моделей;
- дополнительная проверка кода программы при компиляции, в том числе на наличие циклических ссылок;
- существенное упрощение программирования за счет использования стандартных объектов системы Mo.net;
- наличие неявных циклов по времени прогноза, полисным данным и расчету суммарных (по всем полисам) показателей;
- простота создания сложных динамических и стохастических моделей;
- наличие специальных средств преобразования, группирования и слияния полисных данных. Автоматизация описания портфеля модельными точками;
- краткость и наглядность программного кода;
- автоматический вывод результатов расчета, в том числе промежуточных, в табличной форме;
- более мощный дебаггер
- дополнительные возможности отладки, такие как сравнение моделей, анализ используемых переменных и зависящих функций и т.д.

Особенно хотелось бы отметить то, что Mo.net не ограничивает возможностей пользователя; всегда можно делать то, что хочешь, а не то, что позволяет система. Кроме того, фирма OAC Actuaries and Consultants обеспечивает очень высокий уровень поддержки пользователей. При работе с любой, даже хорошо знакомой программой, такой как Word или Excel, периодически возникает вопрос, в котором трудно разобраться самостоятельно. Если программа широко распространена, можно задать вопрос коллеге. Актуарный инструментарий к широко распространенным системам не относится, поэтому задать вопрос коллеге не всегда удастся. Однако по интернету можно задать вопрос специалистам OAC.

Если подготовка ответа на заданный вопрос требует менее 30 минут, ответ предоставляется бесплатно, как правило, в течение нескольких часов. Более того, встроенный архиватор системы Mo.net позволяет приложить к запросу проблемную программу.

Таким образом, дружелюбный интерфейс, высокая скорость счета, простота освоения, простота создания сложных динамических и стохастических моделей, возможность запуска из Excel, обеспеченная Микрософт согласованность с такими системами, как Excel и Access и другие достоинства системы Mo.net делают ее достойной внимания российских актуариев.

И наконец, доступность (цена!). В этом отношении ничего близкого к Mo.net автору неизвестно. При этом низкая стоимость обеспечивается отнюдь не в ущерб качеству, а за счет оригинальных решений разработчиков: использования платформы Microsoft.NET и отказа от больших библиотек стандартных функций. В системе Mo.net можно быстрее написать свои функции, чем найти подходящие в огромных библиотеках да еще и убедиться, что они делают именно то, что нужно.

Желающим ознакомиться с Mo.net предоставляется 28-дневный бесплатный тестовый период пользования полноценной системой. Отметим, что уже в начальном ознакомительном курсе применения Mo.net, доступном в течение тестового периода, на простых примерах, разъясняются такие нетривиальные вопросы, как выполнение множественных прогнозов и реализация стохастических моделей.