Copyright © 2004-10, Dynaplan. All rights reserved

DYNAPLAN

Dynaplan SMIA[1] White Paper

by

Magne Myrtveit[2]

Founder and Managing Director
Dynaplan As

The needs for improved technology

The original spreadsheet was invented by Daniel Briklin in 1978 as a solution to the problems he experienced in his work with spreadsheets based on paper. According to Briklin, the main advantages of the computer based spreadsheet were that tables could be moved freely around to make more space when needed, and that results of formulas could be computed automatically by the computer.

The spreadsheet technology was developed as a pragmatic solution to a practical problem. The pay-off to customers was significant and almost immediate; the price of the spreadsheet software was recovered within a few weeks.

> *Within a decade, North America's financial culture had become a spreadsheet culture. Spreadsheets became the medium, the method, the tool, and the language of all serious financial analysis.(Schrage 2000 p. 38)*

Spreadsheets are use in many different areas and levels of organizations. At the operational level, spreadsheets are used in all areas where numbers are involved. Examples include time sheets for workers; pay roll, bonus and benefit calculations; process planning, reporting, and analysis; balance sheets and income statements; budgeting; consolidation; cost allocation: and much more. Investment decisions can hardly be made without support from calculations made in spreadsheets. And at the strategic level, executives investigate what-if scenarios in their spreadsheets.

The spreadsheet technology also plays an important rhetoric role in the business world. Equipped with impressive spreadsheet models, employees convince their bosses to invest in their ideas. Former competitors implement mergers based on spreadsheet models that back up the believed benefits of the ventures.

The spreadsheets of today are in principle unchanged since the early days of Briklin. Spreadsheets are well suited for creating tabular lay-outs involving numbers and texts, and for performing calculations the data in the tables. Today spreadsheets are used for far more demanding tasks, and significantly more *critical* tasks than what the spreadsheet was designed for. This has led to a situation where modelers in businesses and organization are facing increasingly large problems in managing their

---

[1] SMIA – Structured Modelling Interactive Analysis

[2] Magne Myrtveit has many years experience as software developer, business consultant, founder and managing director. He started his career as a professor in Computer Science at the Universities of Hawaii and the University of Bergen, but soon started his own business. He has been chief architect for three generations of dynamic simulation software; Simtek (form ModellData), Constructor and Studio (from Powersim). In 2002 he founded Dynaplan, where he is leading the development of SMIA.

Copyright © 2004-10, Dynaplan. All rights reserved

spreadsheets. There is a growing body of documentation relating to spreadsheet errors and the problems they cause to businesses. (Panko 1998)

A need for better visualization of structure

Maybe the most significant problem with the spreadsheet is that model structure is interwoven with the layout used for data input and output. Data and structure need to be visualized in very different ways in order to be interpreted clearly by humans. Spreadsheets favor data visualization at the expense of structural clarity. As a consequence, spreadsheet models are hard to understand and difficult to explain to others. The black-box nature of spreadsheets makes it virtually impossible to identify the assumptions and relationships that are included in (and sometimes even more important: excluded from) spreadsheet models. Evidence shows that blindly believing in spreadsheet outputs can be very costly. The business world needs ways to make models more transparent.

A need for better quality assurance

When it comes to quality assurance and maintainability, spreadsheets provide little help to the modeler. The low-level building blocks of a spreadsheet provide great flexibility, at the expense of consistency and clarity. Spreadsheets are composed from cells, much like the memory of a computer. Users can associate names with entries in a spreadsheet, but there is no real concept of a variable. The name of a value or formula is typically entered as a text string in a neighboring cell. Arrays are composed by putting values or formulas next to one another in a row or a column. There is no provision for verifying that array *boundaries* are respected when formulas are entered or edited. Spreadsheets also have a very weak concept of *type*, i.e., the value domain that is acceptable for a variable or a dimension. This, together with the lack of support for measurement *units*, causes the spreadsheet software to accept models that violate basic laws mathematics, physics and logic.

A need to deal with dynamic problems

When applied to long-term problems, spreadsheet models suffer from the lack of support for dynamic feedback. This often leads to significant errors in the simulated results, as feedback chains are eliminated from the problem in order to make the model acceptable to the spreadsheet software. Feedback can have significant impact on model behavior if the forecast horizon is relatively large compared to the time constants in a model.

Furthermore, our mental models are influenced by the tools and methods that we use. Not only does the linearization that is necessary in order to express dynamic problems in a spreadsheet lead to errors in the output, it can also lead to *linear thinking* in the problem solving process. This represents a dangerous side effect of applying spreadsheets modeling to dynamic problems.

Some alternatives to the spreadsheet

The spreadsheet technology dominates the modelling community almost entirely. Alternative technologies seem to struggle to get foothold outside their niche markets. The following reasons may explain this situation:

Copyright © 2004-10, Dynaplan. All rights reserved

- Standardisation—In the global software market, it is quite common that one big winner will emerge in each category. Factors that push in direction include compatibility, market power, revenue, complementary products and services, network effects, etc. (Sterman 2000 ch. 10)
- Ease of use—Spreadsheets are much easier to use than most competing products.
- Scope of use—Spreadsheets are very flexible compared to most competing products, which can be used effectively only on a small subset of the problems that can be addressed by spreadsheets.
- Price and availability—Spreadsheets are relative inexpensive, and at most work places you don't even have to ask for access to spreadsheet software; it is provided to you by default.

The poor visualisation of structure provided by spreadsheets is to some extent solved by some of the graphical modelling tools on the market. Examples include Analytica, Vensim and Powersim Studio. However, these tools have poor or missing functionality for data visualization and manipulation.

Dynamic simulation tools such as Vensim and Powersim Studio provide support modelling of feedback. Unfortunately, these systems cannot be used effectively on systems without feedback. This significantly limits their market. The tools also require extensive training in the field of system dynamics.

There is a range of expert systems that provide powerful functionality for highly trained users. These systems have problems getting out of their niches due to the steep learning curve.

ERP systems provide some modelling capability, but in general the systems are rigid and practically impossible to use for projections other than simple extrapolations of historical data. Standard planning software is simply not flexible enough to fill the need to model alternative scenarios in a constantly changing business environment. (Foster and Kaplan 2001, chapter 7)

Features of Dynaplan SMIA

SMIA is a software program providing integrated and easy-to-use access to state-of-the-art functions for modelling, analysis, and visualisation of data and processes. The software targets business problems ranging from operational planning and reporting to strategic decision making and scenario analysis.

At its core, SMIA is a flexible modelling tool for expressing, visualising and computing cause-and-effect chains. SMIA helps the modeller to cope with structural complexity by providing different levels of abstraction on a model. Model diagrams display variable relationships at any level of detail, ranging from system overview to individual variables.

Data is a central part of modelling, both as input and as output. SMIA supports presentation of data in spreadsheet grids and charts. Multidimensional data can be displayed and manipulated using the mouse, and statistical characteristics such as percentiles, sums and averages are easy to access.

Copyright © 2004-10, Dynaplan. All rights reserved

Risk and opportunity can be expressed using stochastic functions, and the consequences are automatically computed and prepared for visualisation in a variety of ways.

Sensitivity analysis and optimisation can be applied in order to identify leverage points and optimal policies for improving system performance.

SMIA supports both dynamic and non-dynamic models. Dynamic models allow for feedback and time delays. Such models are useful for studying non-linear systems and for investigating strategic problems and other problems with medium to long time horizon.

The system provides access to corporate data. The modelling language is capable of dealing with sparse data, which is a perquisite for efficient import and manipulation of enterprise data.

Models in SMIA are expressed in a high-level language which allows for automatic detection of a range of possible errors. Consistent use of measurement units is enforced by the system. SMIA performs automatic conversion between compatible units in a model, and even between different currencies such as $ and €.

Quality assurance and maintainability is facilitated through SMIA's strict enforcement of mathematical and logical consistency within and across model equations. Graphical visualisation of complex cause-and-effect structures as well as multidimensional data helps the user in understanding and communicating how a model is built up, what it performs, and how it can be modified without breaking its logic.

Characteristics of Dynaplan SMIA

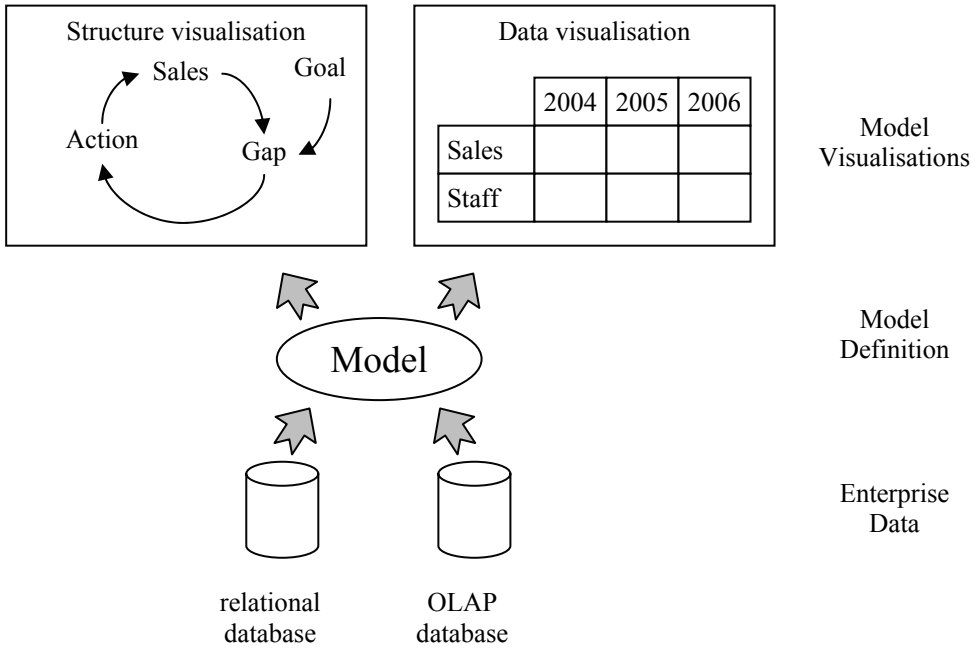
Dynaplan SMIA provides a feature set that is a superset of spreadsheet and system dynamics technologies. The system is designed with ease-of-use in mind. The number of concepts that the user needs to master is kept to a minimum. The user interface automatically unfolds whenever the user makes use of new features of the software. SMIA takes a broad perspective on the task of modelling, as it includes support for a wide range of different kinds of models:

- non-dynamic models—Users can express cause-and-effect relationships in a graphical modelling language and/or using a spreadsheet metaphor.
- dynamic models—Such models involve feedback and delays. Users can enable the system dynamics functionality to express levels and flows and determine the behaviour over time for dynamic systems.
- stochastic models—The user can perform calculations involving uncertainty, producing distributions based on Monte Carlo simulation
- statistical models—The user can analyse data and compute percentiles, averages, deviations, trends, etc.
- object models—Models can be created as a collection of re-usable objects along the lines of object-oriented programming.
- database models—Data from relational databases and OLAP databases can be imported and manipulated (NOTE: to be implemented)

Based on object-oriented design the system separates between the conceptual models and the visualisations of the models. This approach leads to great flexibility. In one

Copyright © 2004-10, Dynaplan. All rights reserved

diagram a model can be visualised graphically as a chain of causes and effects. At the same time the model can be displayed in one or more spreadsheet grids containing multidimensional data.



The above figure illustrates that enterprise data[3] can be imported into a model. It is also possible to make models that do not make use of external data. The model definition can be edited through the structural view or through the data view, or both. Results can be displayed in many different views, including causal loop diagram (top left) and spreadsheet view (top right). Many different graphical views (charts) are also provided.

Feature summary

The sections below describe a subset of SMIA's characteristics. The list is divided into three:

user interface—how the user works with models,

document—what a model is like, and

analyses—results that the user can produce.

User interface features

SMIA is designed to follow the standards and conventions of contemporary user interface design. The product is build using cross platform technology, which makes it relatively easy to create versions for Linux and Mac, as well as the current version which is developed for Microsoft Windows.

Multi-level undo and redo—Changes to a document can be undone to arbitrary level, and undone commands can be redone.

Multi-selection—Users can select an arbitrary number of graphical objects to perform actions using mouse or keyboard.

[3] Import from databases is currently done via flat files or spreadsheets.

Copyright © 2004-10, Dynaplan. All rights reserved

Modeless property pages—Properties of objects in the current selection can be observed and changed through a modeless property page concept. (Modeless property pages can stay open while editing the model.)

Context menus—Click with the right mouse button opens up a pop-up menu with actions that are relevant for the current selection.

Drag and drop—Objects can be moved and copied using a drag-and-drop operation with the mouse. It is possible to drag objects to and from other applications, such as word processors and spreadsheets.

Localisation—The software is enabled for language translation. The current version is in English, German and Norwegian Nynorsk. New versions can be created using a translator tool

Document features

Below is a list of some of the important features that can help users visualise models and ensure their quality.

Cause-and-effect visualisation—Variables (and other elements) of a model can be visualised in diagrams using boxes and arrows.

Spreadsheet visualisation—Variables can be visualised using a spreadsheet metaphor. Multidimensional data can be inspected. Drilldown, grouping, totals and statistical functions can be applied automatically to the data.

Hierarchy—Variables can be put into hierarchies. This helps the user organise models into systems and sub systems, for example. The hierarchy can be built using drag and drop.

Scalars and arrays—A scalar is a variable that holds a single value. Arrays have elements that can be arranged in one or more dimensions. A one-dimensional array is also called a vector. A matrix is a two-dimensional array. Explicit support for arrays opens up for better quality assurance as well as easy-to-use statistical analysis and visualisation of data. The system can automatically detect out-of-bounds references to arrays, and inform the user. (This is one of the most common mistakes that go unnoticed in spreadsheets.)

Compact and sparse arrays—A compact array has values for all its elements, whereas a sparse array can have empty slots. Sparse arrays are necessary for efficient representation of data in corporate databases (ERP systems). They are also useful for expressing commonly used array dimensions in the real world. An example is the way bookkeeping is arranged into accounts. Each account is identified by a number (an integer). There are typically large jumps in the numbering sequence for accounts. The account model (as opposed to the key figure model) in SAP systems is a good example of a sparse dimension. SMIA can represent both the key figure model and the account model efficiently.

Self-referencing definitions—SMIA allows array definitions to be self-referencing. This makes it possible to define a normal spreadsheet as a special case of a sparse array in SMIA. Another benefit of self-referencing definitions is that a range of commonly used spreadsheet structures can be expressed in a very compact, robust (little risk for error), and maintainable way. Here is an example:

Copyright © 2004-10, Dynaplan. All rights reserved

Spreadsheet

	A	B	C
1	2004	100	=B1*10%
2	2005	=B1+C1	=B2*10%
3	2006	=B2+C2	=B3*10%
4	2007	=B3+C3	=B4*10%

SMIA

y	Account	Interest
2004	100	=Account * 10%
2005	=Account(y-1) + Interest(y-1)	
2006		
2007		

The spreadsheet (left) repeats similar formulas in the columns B and C, except for in the cell B1. SMIA uses three formulas only: one for the initial value of Account, one for the remaining values of Account, and one for Interest. Note also that SMIA refers to other values by name as opposed to location (row and column number).

Measurement units and types—Variable values have predefined or user-defined types and units. The predefined types include real numbers, integers, text, and logical values. The predefined units include the SI units metre, second, Celsius, etc. The US and British customary units are also included as built-in measurements.

Support for types and units makes models more readable since values are input and output with the correct unit. As an example, a variable representing 12 345.00 euro is output as € 12 345.00. If the value is a flow, this information will be provided as well, for example like this: 12 345.00 €/yr. See also *contents colouring*, below.

Types and units also make models more reliable, as the system will automatically ensure that the laws of mathematics and logic are obeyed by the model. It is, for example, not possible to add two values unless they have compatible units. SMIA automatically converts between units, if necessary. The expression 10 m + 5 feet will evaluate correctly, for example. The same is true for 10 m/s + 36 km/hr.

The *mixed* data type corresponds to "variants" in other languages. This type makes it possible to represent spreadsheets as normal SMIA variables, and it makes it possible to import OLAP data with mixed units, such as key figures with embedded currency information.

Currencies—A currency is a special kind of unit. The exchange rates for currencies can be determined as constants or as functions over time. Historical and current exchange rates can be downloaded directly from the European Central Bank over the Internet. Different currencies can be mixed freely in a model, and the relevant exchange rates will be applied depending on currencies involved and the timing of the currency transfer. As an example, it is possible to write 10 EUR + 20 USD and store the result in a variable defined as EUR, USD or any other currency.

Contents colouring—The type and unit of an object holds the information about the kind of "material" that is held by the object. An automatic, customizable, colouring system can be enabled to display objects with colours representing type of contents. In a model with many variables, this can greatly enhance readability. As an example, a model with products, money, costs/prices and time can be coloured using four different colours. Varying saturation and darkness is used to represent flows as opposed to stocks.

Copyright © 2004-10, Dynaplan. All rights reserved

Multiple languages—Users can add languages to a document. Documents are edited in a language selected by the user. Texts can be translated using a separate dialog where objects are listed along with their translations.

Values are formatted according to the conventions of the selected language. (Choice of decimal separator, names of months, days, and currencies, etc.)

Analysis features

SMIA include broad range of analyses. Relevant analyses are performed based on the features used in a model. Options that are relevant for a given model will be presented to the user. Options that are not relevant will remain hidden.

Calculation—This mode corresponds to standard spreadsheet behaviour, where the cause and effect relationships between values are used to evaluate the model. An example:

Income = Price * Sales
Price = 10 EUR/item
Sales = 15 item

Risk analysis—This mode involves randomness, and it is automatically enabled if one or more variables use a random function. The results will be calculated using Monte Carlo simulation, or other techniques that the user can select from. We can make the above example into a stochastic model by making Sales a random variable:

Income = Price * Sales
Price = 10 EUR/item
Sales = random(10 items, 19 items)

The above model will generate uniformly distributed samples for Sales in the range between 10 and 19 items. The distribution of values produces for the dependent variables (here: Income) will be possible to study directly in the user interface, measures as percentiles, averages, standard deviations, etc.

Simulation—This mode opens up for simulation (over time or over another dimension). Dynamic simulation is useful for expressing feedback that is involved in the problem that is modelled. Simulation mode is automatically enabled in models that make use of the implicit simulation dimension (the *horizon*), for example when a variable is defined as a flow. Here is an example:

Account = **stock** 1000 EUR **inflow** Interest
Interest = Account * 'Interest Rate'
Interest Rate = 5%/year
Horizon = date(2004) **to** date(2010) **step** 1 year

Here Account is a stock variable (an accumulator) which is initialized to 1000 €. In each time step (typically one year) Interest is added to the account through an inflow. The Horizon is a predefined type, determining the start, stop and step of the simulation. Here we start in 2004 and stop in 2010, with an increment of 1 year between each simulation period.

Copyright © 2004-10, Dynaplan. All rights reserved

Dynamic risk analysis—This mode is enables when a model contains both feedback and randomness. In our example above, we can for example make Interest Rate into a random variable, like this:

Interest Rate = normal(5, 3) %/year

Sensitivity analysis—Here the user can study the effect of systematic changes to one or more independent variables. Output from sensitivity analysis is typically presented using scatter graphs or tables, where X is an independent variable and Y a dependant variable.

Interest Rate = **parameter** 3%/year **to** 10%/year **step** 1%/year

Optimisation—Here the modeller specifies an objective which is automatically approached by the system through a search algorithm. In addition to the objective, the modeller will specify which parameters the system can alter in order to look for the optimal solution.

Optimum = **maximize** Account

Stochastic optimisation—Here the optimisation is used on a model involving random functions. The objective is expressed as characteristics of the distributions generated for dependant variables. As an example, the modeller can maximize a given percentile of a variable such as Profit.

Dynamic optimisation—Here optimisation is used on a model involving time. The objective can be based on final values for the simulation, or characteristics of the time series generated by the simulation.

Dynamic, stochastic optimisation—This mode can be used to optimise policies in an uncertain environment. This analysis is enabled when the model contains both feedback and random functions.

Scenario analysis—A scenario consists of a coherent set of inputs (assumptions and decisions) together with the corresponding set of outputs when fed into a given model. Scenarios can be created, documented, updated and compared in order to illustrate and evaluate alternative strategies and possible future developments. The built-in scenario is a key feature in strategic and operational planning.

Rolling forecast—This feature reduces the amount of manual work involved in performing periodic planning activities. It takes just a click on a button to prepare a model of any size to the next planning horizon, making the model ready for input of current data and revised assumptions for the future.

References

Foster, Riechar, and Sara Kaplan. 2001. *Creative Destruction: From build-to-last to build-to-perform*.

Panko, Raymond R. 1998. What We Know About Spreadsheet Errors. *Journal of End User Computing's* 10 (2):15-21.

Schrage, Michael. 2000. *Serious Play. How the World's Best Companies Simulate to Innovate*. Boston: Harvard Business School Press.

Sterman, John D. 2000. *Business dynamics : systems thinking and modeling for a complex world*. Boston: Irwin McGraw Hill.