Table of Contents

Go Back

"Instantaneous processes" – A practical requirement of System Dynamics !?

Dipl.-Ing. Claudia Dingethal
University of Bergen/Norway
Wallaustrasse 58, 55118 Mainz/Germany
Phone: +49 (0) 163 / 47 567 49
E-mail: dingethal@gmx.de

Abstract

In many of the systems that are subject to System Dynamics modelling in client projects, instantaneous information processes take place. The purpose of such processes is to conduct a numerical analysis (incl. optimization) to support decision making. The time-span on which these processes take place is insignificant compared to the time span on which we investigate the system at hand. The implication is that such processes are considered to take place instantaneously, i.e. without the passing of time, i.e. at distinct points in time. Most modelling and simulation software, developed within the context of system dynamics, have not been designed to incorporate iterative numerical processes of this kind. This paper intends to open up a debate about the necessity, usefulness and possibilities of incorporating instantaneous processes into System Dynamics models. It presents first research results and possible areas of application in which iterative information processes play a significant role.

Keywords: Iteration, Instantaneous processes, asset lifecycle, decision theory, optimisation

1 Introduction

1.1 Problem Formulation

Since its first steps in the late 60’s System Dynamics has proven its superiority in analysing complex, dynamic and non-linear systems. The awareness and the use of System Dynamics in business environments have increased over the last years.[1] Managers, today, appreciate to a high extent the possibility of clearly clarifying main interrelations in their decision environment as well as explicitly considering feedbacks and time delays. Figure 1 provides an overview of areas of System Dynamics application in the business environment.

	Case by Case	Ongoing Basis
Decision Support	Strategic Decision Making	Planning and Budgeting
Communication	Value Communication	Interactive Learning Environments

Figure 1: Examples of areas of System Dynamics Applications[2]

Modellers encounter more and more real situations and clients with high-stakes questions that are nearly impossible to answer by using only the fundamental concept of System Dynamics.[3] In this paper we refer specifically to the necessity of including instantaneous decision processes into continuous simulation models, that means:

Decision makers, in the real world, base their decisions on decision models that represent the slice of reality relevant to their decision. As a result of this "information processing" they implement the conclusion (= decision) into the real world (see upper part of Figure 2, page 3). That means, the decision maker applies a certain decision rule at a distinct point in time that is relevant for a time span that is much longer than the time span of the decision process itself. In other words, the time span in which the decision process takes place is insignificant compared to the time horizon that is influenced by the decision.

The inclusion of such decision processes into continues simulation models is a challenge faced by modellers (see Figure 2). Simulation models may have to incorporate the decision process of the decision maker. In that way the model user is able to conduct strategic scenario analyses in a continuous, dynamic simulation environment which processes important and complex decision processes automatically. In other words, the real decision making process at distinct points in time is part of the continuous model.

---

[1] see project references at the home page of the three main software vendors in System Dynamics: High Performance Systems (www.hps-inc.com), Powersim (www.powersim.de), Ventana Systems (www.ventanasystemsinc.com).
[2] Figure adopted from Powersim webpage.
[3] The term "hybrid modeling" plays an increasing role in System Dynamics, see for instance: Love, G. (2001), Schwarz, R. (2003), Levin, T./Levin, I. (2003), Osgood, N. / Kaufmann, G. (2003).

The decision making process can either deal with problems that can be solved straightforward or with problems that can be solved only by iteration. When speaking about straightforward, we refer to decision processes where the solution to a certain problem can be found in one step. The mathematical representation of such processes is $x = f(x)$. Yet, in some cases a final decision can only be found solving complex, mathematical problems. By repeating (or iterating) a recursive procedure that gives successive approximations or an exact (resp. optimal) result the final decision will be found. Most modeling and simulation software, developed within the context of system dynamics, have not been designed to incorporate such numerical processes (including optimization).

1.2 Research Goals

The necessity to include iterative processes when applying System Dynamics in the business environment has already been mentioned by BOB EBERLEIN and BILL STEINHURST in 1997. Yet, the technical realisation and practical usefulness has so far a low presence in scientific publications.

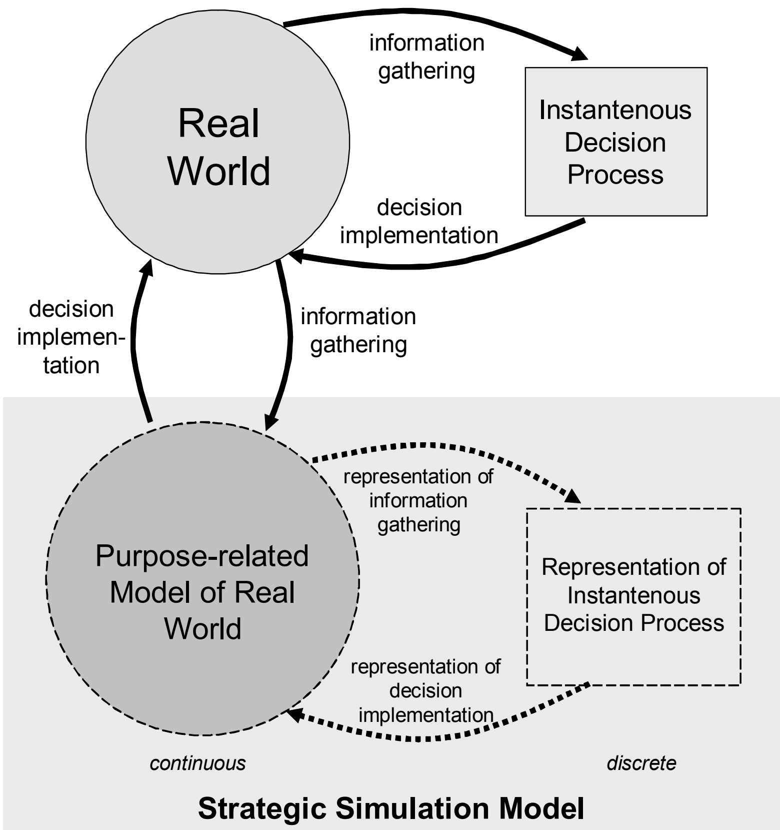


Figure 2: Research subject

The main objective of this project is to identify and discuss a number of cases from the business world, where it is necessary to include instantaneous processes into a continuous simulation model. The research will focus on three fields: asset lifecycle, decision theory and optimisation. Processes identified in the research, will be formulated using constants and auxiliaries in a first step, if possible. This follows the glas box philosophy of system dynamics. Technological enhancements of System Dynamics software tools provide the opportunity to formulate and incorporate such processes with help of a Visual Basic Scripting Function into simulation models. In that way, a much more flexible possibility of using instantaneous processes is available. Therefore, all processes will be transferred into Visual Basic Scripting functions in such a way that they can be used as generic structures in System Dynamics models.

As the inclusion of instantaneous processes is an enhancement to the fundamental concept of system dynamics, it is also necessary to analyse the theoretical implications of this approach.

2 Theoretical Implications

2.1 System Dynamics and the power of iteration

System Dynamics, the methodology used in this thesis, is a continuous simulation technique. VDI-Guideline 3633 defines a "Simulation … [as] the replication of a system with its dynamic processes in a model that can be used for experiments in order to reach conclusions, which can be transferred into reality."[4] In other words, simulation is the work of a model that replicates a real system. The great advantage of simulations is that the model can be manipulated in a way that would be impossible, too dangerous or too expensive in a real system. The behaviour of the model can be tested and conclusions about the behaviour of the real system can be drawn.[5]

The continuous simulation method System Dynamics has been applied in many domains and has proven as an efficient mean in the analysis of complex, dynamic systems. By emphasising the holistic perspective, System Dynamics helps us understanding complex, dynamic and continuous systems and explains how structure governs behaviour. Stock and flow diagrams as well as causal loop diagrams provide a visible description of the relationship between the variables identified as significant in the formation of the systems' behaviour.[6]

In the introductory part we introduced the necessity of incorporating complex instantaneous processes into System Dynamics models. Most complex decision process can only be solved by iteration. The term iteration (lat.: repetition, repeated application) describes in general a step-wise approach for finding a certain result, whereas the same procedure is used several times.[7] In other words: iteration is any method for arriving at a result by repeating a recursive procedure that gives successive approximations[8] or an exact (resp. optimal) result. DUBUC describes iteration as “a royal way for a majority of questions in numerical analysis and in optimisation.”[9].

An iteration formula describes the procedure of calculation to be iterated. Based on a given formula

$x = \varphi(x)$ and an initial value $x_0 = I$

---

[4] VDI Guideline 3633.
[5] see Mertens, P. (1982, p. 1).
[6] For further information about System Dynamics, please refer to Forrester, J.W. (1961) and Sterman, J.D. (2000).
[7] see Brockhaus Enzyklopädie (1990, p.48).
[8] see Glenn, J.A. (1984, p. 105).
[9] Translated from Liedl, R./Reich, E./Targonski,G. (1985, p. 2).

a certain sequence of $\langle x_n \rangle$ is defined by using the mathematical instruction

$$x_1 = \varphi(x_0)$$
$$x_2 = \varphi(x_1)$$
$$x_3 = \varphi(x_2)$$
$$\vdots$$
$$x_n = \varphi(x_{n-1}) .$$

In the field of System Dynamics the term iteration can so far be found in two contexts. On the one hand it is referred to the modelling process itself: "A well developed System Dynamics model should get through multiple rounds of revision and evaluation."[10] System boundaries, level of aggregation or detailed formulations might be changed.[11] The iterative process may continue as long as the model fails to satisfy some evaluation criterion.[12] In this case the intermittent outcomes of the iteration process $\langle x_i \rangle$ are various versions of a model at various stages of its development.

On the other hand, the subsequent simulation is itself an iterative process. The same structure is applied to the state of the model so that the state of the system at time t depends on the state of the system in time t-1. The resulting state trajectory of the model constitutes its behaviour.

In this project we refer to a third context in which the term iteration is applied in the field of System Dynamics: an iterative process that takes place at distinct points in time. This iterative process is a representation of a decision making process in reality. The implication is that such processes are considered to take place instantaneously as the time span on which these processes take place is insignificant compared to the time span on which we investigate the system at hand.

2.2 Technical Realisation

This paragraph introduces three approaches of technically realisation:

- explicitly modelling the structure of an iteration process
- using available software programs
- using program scripts inside System Dynamics Software

Explicit modelling

The option of explicitly modelling the structure of the complex, instantaneous decision process follows the glass box approach of System Dynamics. Using auxiliaries and constants provides a visible representation of the process. As the level-rate concept is not included in the structure, no time delays occur.

This approach allows grasping and validating the general relationships between the elements of the decision process without having to analyse the

---

[10] Homer, J.B. (1996, p. 2).
[11] see Homer, J.B. (1996, pp. 2-3).
[12] see Randers, J. (1980, p. 118).

mathematical details. In that way it provides a basis for managers and modellers to reach a common understanding of the underlying structure.[13]

However, modelling the process explicitly is an inflexible solution regarding two aspects. First, the structure of the process has to contain explicitly the maximum number of iterations. It is not possible to formulate a condition which exits the iteration process, i.e. which neglects the computation of certain variables, respectively auxiliaries. That means, even though a solution or an acceptable good enough solution was found after a certain number of iterations, all modelled iteration steps are processed in every point in simulation time, i.e. all auxiliaries are computed, although not necessary. This slows down the simulation speed, leading to a lower performance of the simulation.

Secondly, the solution is inflexible concerning changes in ranges when working with arrays (multi-dimensional variables). The number of decision steps can depend on the number of elements in a range. Thus, changes in the number of elements in a range require changes in the model structure.[14]

In addition to the inflexibility issue the structure of the decision process can easily become unclear when modelling complex processes. In that way it could lead to low acceptance and a refusal of the model. If the structure of the decision process is not a major concern for understanding the relationships in the main model, it might become useful hiding the structure. System Dynamics software tools provide several possibilities of hiding structure, e.g. hierarchical models.

Advantages	Disadvantages
Following the glass box approach	Inflexibility concerning end of iteration process
Structural visualisation of relationships of the iteration process	Inflexibility concerning changes in the number of elements in a range
No knowledge of any programming language required	Becomes too complex and unclear in large iteration processes

Table 1: Advantages and Disadvantages of explicitly modelling complex decision processes

Using available software programs

Most System Dynamics modelling tools provide a data interface. This data interface can be utilised for the purpose of this project in the following way: Information, respectively the value of variables, required for the decision making process, are exported to a data processing software that conducts the decision process. Instantaneously, i.e. in the same point in simulation time, the result of this process is imported by the System Dynamics software in order to forward the

[13] There is nothing worse than a user that does not trust the results of the simulation model, even though he should always question the simulation results.

[14] For iteration processes that means: Adding an element requires copying an iteration step, i.e. variables forming one iteration step, renaming the copied variables and adapting the equations of the variables that connect two iteration steps. Deleting an element requires deleting an iteration step. In both cases, the equations of the follow-up variables, i.e. variables connected to the result of the iteration process, have to be adapted, as well.

simulation.[15] It might not always be possible to address the data processing software, directly. In these cases external files (e.g. spreadsheets) might work as transmitter between System Dynamics software and the software conducting the iteration process.[16]

By taking advantage of the data interface for connecting System Dynamics models and software programs that are able to conduct the complex decision making process relevant for the purpose of the model, the need for modelling explicitly an iteration processes in System Dynamics Software as shown in 0 is not given.[17] Iteration algorithms which have already been created and tested might be used, decreasing time and effort for model creation and validation.

This approach is contradicting to the glass box approach of System Dynamics. The iteration process is conducted outside the System Dynamics software, running in the background of the simulation. Consequently, this part of the model becomes a black box. Moreover, the user might have to invest into software programs that can conduct the necessary iteration processes. This increases the project costs and might lead to a complete rejection of the project.

The data interface is used according to the pre-specified points in time, resp. the exporting/importing interval. That means, the points in time at which the decision process is conducted do not depend on the state of the system, resp. its dynamic behaviour. At these pre-specified points in time necessary data are exported and the result is imported, instantaneously. A frequent data transfer can slow down the simulation speed, leading to long simulation run times.[18] **Fehler! Verweisquelle konnte nicht gefunden werden.** summarises advantages and disadvantages of using available software programs.

Advantages	Disadvantages
No modelling necessity inside System Dynamics Software.	Black Box approach
Usage of available software programs, i.e. available knowledge.	Data interface is used twice at every pre-specified point in time.
	Data export and import only at pre-specified points in time.
	Long simulation run times.
	Software investment costs.

Table 2: Advantages and Disadvantages of using available software programs

[15] This approach is not working with DDE (Dynamic Data Exchange) interfaces as they have an inherent delay of one time step. The COM (Component Object Model) is an appropriate interface for exporting, processing and importing data in one time step.
[16] Example: Powersim → Excel → Iteration processing software → Excel → Powersim.
[17] Several add-ins to Microsoft Excel are available which support iterative processes, specificially optimisation, inside the spreadsheet. Microsoft itself provides the Excel Solver®. It can solve linear and integer problems by using the simplex method and the Branch-And-Bound-Method. Other providers, e.g. Lindo (www.lindo.com), offer add-ins to Microsoft Excel which are able to solve more complex optimisation problems (*LOVE* embedded a linear programming algorithm using Lingo 8 (a product of Lindo) in a Powersim Studio Simulation Model; see Love, G. (2001)).
[18] see Love, G. (2001, p. 15)

Program Scripts inside System Dynamics Software

As using the data interface frequently slows down the simulation speed, it would be of high benefit conducting the decision process inside the System Dynamics software tool. Powersim Studio 2003 provides the possibility of writing a Visual Basic Scripting (VBS) code directly into a variable of the model.

In that way the data interface to additional software programs is not used and, therefore, does not affect the performance of the simulation. In comparison to explicitly modelling the complex decision process (see 0), the iteration process can be designed flexible towards changes in ranges and can be stopped as soon as a final solution is found. Moreover, complex and large iteration processes can be incorporated easily. In addition, the points in time at which a decision process is conducted can be formulated depending on the state of the system.

This approach requires knowledge about the programming language. The program code is written directly inside a variable. Consequently, the structure of the iteration process is not visualized in the System Dynamics Model and, therefore, is contradicting to the glass box approach of System Dynamics.

Advantages	Disadvantages
No usage of data interface	Black Box approach
Flexibility concerning end of iteration process	Requires knowledge of programming language
Flexibility concerning changes in the number of elements in a range	
Complex iteration processes can be incorporated easily	
Flexible points in time for complex decision process	

Table 3: Advantages and Disadvantages of programming inside System Dynamics Software

Implementation in this project

In general, it depends on the specific circumstances of every single project, which one of the options described above will be used. Effort and benefit need to be taken into consideration. Moreover, experiences from former projects and available resources might influence the choice.

In this project we will, in a first step and where possible, formulate the iteration process using constants and auxiliaries. This follows the glass box philosophy of System Dynamics methodology. In a second step we will utilise the technological enhancements regarding incorporating VBS. All iteration processes will be transferred into VBS in such a way that they can be used as generic structures in System Dynamics models dealing with the same problem. By providing both options, explicit modelling and incorporating VBS, a modeller using these iteration processes in his own model can choose which option is more suitable for the project at hand.

2.3 The modelling process

The modelling process is an iterative process which continues as long as the model fails to satisfy some evaluation criterion.[19] Consequently, modelling is a feedback process, not a linear sequence of steps. Sterman identifies five steps of the modelling process: problem articulation, formulation of dynamic hypothesis, formulation of simulation model, testing, policy design and evaluation.[20] Feedback can occur from any step to any other step. Figure 3 incorporates the creation of instantaneous processes into this five step modelling process.

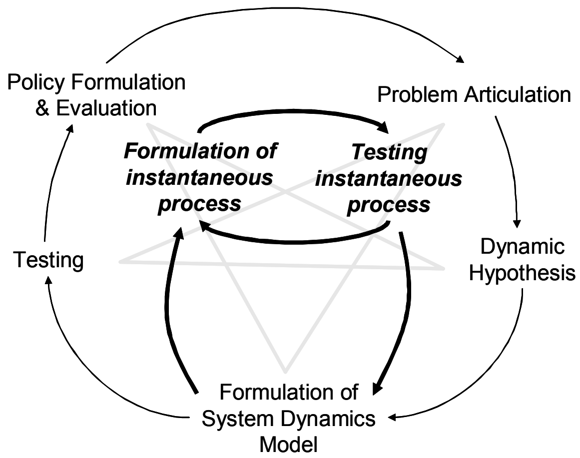


Figure 3: The modelling process

As a first step the problem which should be solved with help of the simulation has to be defined. Specifying a clear purpose is the basis for a successful modelling study. After identifying the problem a dynamic hypothesis has to be formulated that explains the problematic behaviour. This includes defining system boundaries as well as endogenous and exogenous variables. Once an initial dynamic hypothesis has been developed it can be started to create the System Dynamics model.

From the authors experience it can be suggested to build in a first step the model on a high aggregated level that does not require complex decision processes. This model can be used to explain aggregated behaviour, conduct first policy analysis and conclude what can be learned from this model. Analysis might already generate insights into the dynamics of the modelled system and might provide first solutions to the problem at hand.

We suggest creating, testing and, if necessary, refining the instantaneous process separately, in order to build up confidence (which is an iterative process in itself). Afterwards it can be implemented into the System Dynamics model. After testing the successful implementation the model user can analyse the simulation results and gain further insides in order to solve the formulated problem.

[19] see Randers, J. (1980, p. 118).
[20] see Sterman, J.D. (2000, pp. 86-87).

2.4 Verification and Validation

Model verification and validation is a crucial procedure in all modelling techniques and methodologies. Especially in commercial models modellers have a duty of care that the model is "correct", respectively "valid", as the client otherwise may be led to erroneous conclusions.[21] It is a „distributed and prolonged“[22] process which takes place throughout the whole modelling process.

The concept of incorporating complex, instantaneous decision processes into System Dynamics models leads to an extension of the current "standards" of model verification and validation. A three-level-testing-hierarchy is suggested:

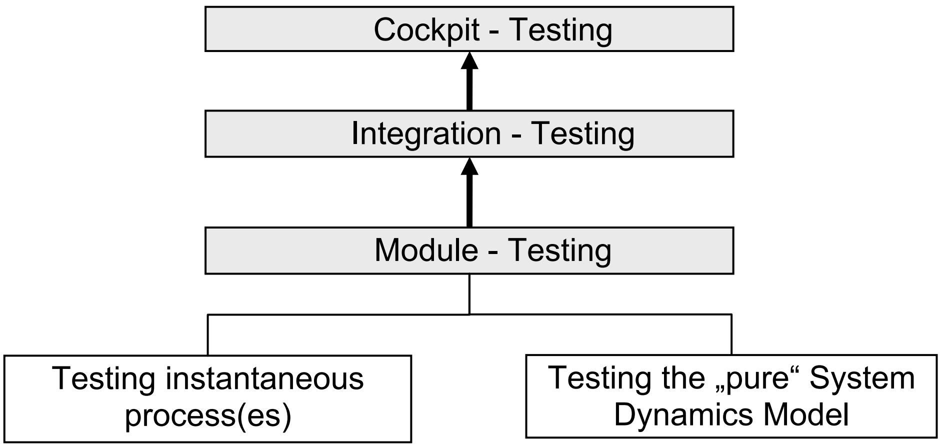


Figure 4: Testing hierarchy

Module Testing

Module testing forms the first level in the hierarchy. The overall objective of this level is to find bugs in logic and algorithms in both kinds of modules; the formulation of the instantaneous process as well as the System Dynamics model. It is suggested, to build and test them in a first step separately. In that way, confidence can be build first into the separate modules, before integrating them.

There are two main categories of tests that can be distinguished: static and dynamic tests.[23] Static tests analyse the program code without executing the program. It is rather analysed by inspection, reviews and walkthroughs. Static tests are also called direct structure tests. Dynamic tests analyse the program by execution under different conditions. Herein specific input values are defined and the result of the program is analysed. Dynamic tests subsume structure-oriented behaviour tests and behaviour validity tests. As there is a wide variety of literature dealing with testing program codes on the one hand and testing System Dynamics Models on the other hand we will not go into details about Module Testing in this paper.[24]

---

[21] see Coyle, G./Exelby, D. (2000, p. 27).
[22] Barlas, Y. (1996, p. 2).
[23] see for instance Zimmermann, P.A. (1987, p. 63), Balzert, H. (1999, p. 509).
[24] For further information, please refer to Balci, O. (2003), Balci, O. (1998), Barlas, Y. (1996), Sterman,, J.D. (2000).

Integration Testing

After module testing the next level is applied - integration testing. The intention is to find bugs in the interfaces between the modules. Although all modules have been tested separately, problems can occur when they work together. Specifically, when formulating the decision process using VBS, it has to be tested whether all functions are working correctly. A step-wise integration is suggested, whereby the VBSs are implemented one after another. The following tests are summarized from the author's experience.

As well as in the separate tests of the VBS and the System Dynamics model, static and dynamic tests can be distinguished. Static tests include dimensional consistency, unit testing and time usage; dynamic tests include the timestep test .

Dimensional consistency: the declaration area of variables inside Powersim variables is different from the one used in the "normal" VBS language. It has to be ensured that the declaration of the dimensions for all input variables and the dimensions of the output variable are correct. The output dimension has to be determined by the modeller himself.

Unit testing: In general units of auxiliaries and flows are calculated automatically by Powersim. Yet, this is not true for auxiliaries and flows calculated by a VBS. Units of the input variables are not considered and, therefore, the unit of the output is not calculated. Consequently, the correct unit has to be determined by the modeller at the end of the script.

Time usage: When working with time units Powersim translates the time value into seconds, e.g. 1 hour results in a value of 3600. The modeller has to be aware of this peculiarity and has to ensure that the VBS is using the correct value.

Timestep test: It depends on the modellers judgment and experience in what frequency, so in what time interval, the implemented decision process needs to be conducted. If this time interval is smaller than the timestep found in the separate System Dynamics module testing, than the timestep of the System Dynamics model needs to be reduced. If, otherwise, the iteration should be conducted less often than the found timestep, the variable containing the iteration process should include a time- or state-dependent IF-function.

Cockpit Testing

As most commercial System Dynamics models are provided with a user cockpit, the top level requires cockpit testing. The objective of this test is to determine, on the one hand, whether the cockpit meets the needs of the customer in respect to knowledge transfer and scenario analysis ("cockpit validation") and, on the other hand, it should also test the correctness of navigation and presentation ("cockpit verification").

A useful cockpit requires a thoughtful concept which needs to be discussed with the customer in the very first stages of the design phase. The discussions can be supported using the technique of SADT (Systems Analysis and Design

Technique).[25] SADT helps to define input, output and decision influencing (environmental) elements that should be included in the cockpit. Input variables can be influenced by the end-user in order to run and analyse different scenarios. The decision of the end-user might be based on certain environmental aspects, which have to be provided in the cockpit as well. The output is calculated based on the input, the environmental aspects and the underlying model and should support analysing scenarios.

The objective should be developing a clear, concise and consistent design.[26] An easy-to-use cockpit is characterised by convenience, clearness, regularity, uniformity and familiarity. Elements of interaction (controls, slide bars, switches, etc.) should be used in a conservative way, without using many different colours or fonds.[27] Cockpits lacking these aspects are in general perceived as being unpleasant and bulky.[28]

During the design phase formal aspects have to be considered as well which should be tested at the end. This refers for instance to the correct definition of hyperlinks and to input verification. Only valid input should be allowed, otherwise a comment should be provided, which elucidates the user about the wrong input. This aspect can be tested by defining and executing special input scenarios and providing an appropriate message box.[29]

---

[25] see Krallmann, H. (1994, p. 60-64).
[26] see Rakitin, S.R. (1997, p.224).
[27] see Balzert, H. (1999, p. 716).
[28] see Kerninghan, B.W./ Pike, R. (2000, p. 134).
[29] see Zimmermann, P.A. (1987, p. 38).

3 Instantaneous process in Action – possible areas of application

3.1 Asset Lifecycle

Investment/disinvestment decisions and their rationality have been a focus of investigation among System Dynamics modellers.[30] Many System Dynamics models succeed in describing how investments actually are made. Mostly, it is figured out that the considered time horizons are too short and/or just current conditions are taken into consideration, as for instance, the current gap between the desired and actual stock of assets and the currently perceived profitability of the business.[31]

Yet, when applying System Dynamics in the business environment, in some cases, it is important to incorporate the asset lifecycle the way it should be handled, not how it actually is handled. Specifically in the field of planning and budgeting a more "precise" representation of reality becomes necessary. That means, decision models, i.e. information processes, as described in scientific literature have to be part of the model. In this paragraph we introduce the following instantaneous processes over the lifecycle of an asset: investment decision based on internal rate of return, finite declining depreciation and value related disinvestment.

Investment decision based on Internal rate of Return

The process of investment decision making has been analysed in financial literature extensively.[32] Choosing investment optimally requires formulating and solving a complex, stochastic, dynamic problem. Managers must have knowledge about future developments and behaviour with regard to costs and demand facing the firm, in addition to all variables and other actors in the system. Moreover they need to have the cognitive capability and the time to solve the resulting problem. None of these conditions is met in reality. In practice, investment models are based on severe simplifying assumptions to render the problem tractable.[33] The most widely used models are net present value (NPV) and internal rate of return (IRR) calculation in order to represent the profitability of an investment.[34]

The IRR measures the rate of profitability. IRR is the discount rate that makes the present value of cash flows (revenues minus expenses) equal to the initial investment. In simple terms, it is the discount rate that makes the NPV of an investment equal to zero. If the IRR is greater or equal than a specified constant discount rate, the investment is profitable and will be realised.

Mathematically this can be expressed as follows:

Formula I: $$0 = -I_o + \sum_{t=1}^{n} \frac{(R_t - E_t)}{(1+i)^t}$$

---

[30] Mentioning all publications here is impossible. Exemplarily the following publications should be mentioned:

[31] see as an example: Dingethal, C. (2000).

[32] see for instance Brealey, R.A./Myers, S.C. (2000); Perridon, L./Steiner, M. (1997).

[33] see Sterman (2000, p. 599).

[34] see Brealey, R.A./Myers, S.C. (2000, p. 93).

whereas I_0 = Investment at time 0
R_t = Revenues at time t
E_t = Expenses at time t
i = internal rate of return[35]

In general there is no closed-form solution for IRR calculation where T is larger than 5.[36] One must find the solution iteratively.[37] Based on an initial interest rate the NPV is calculated. In case that the result of the NPV calculation is positive, the interest rate was too low and has to be increased. In case that the result of the NPV calculation is negative, the interest rate was too high and has to be reduced. By repeating this process several times an appropriate approximation for the IRR is reached.

When incorporating this structure using auxiliaries and constants a fixed number of iteration steps needs to be specified, assuming that the calculated IRR after this number of iterations is close enough in order to have a sufficient good approximation. The structure in itself is transparent and can easily be understood by clients. Using the possibility of VBS allows stopping the iteration procedure as soon as a sufficient good enough approximation is reached. On the one hand this increases simulation speed and ensures, on the other hand an acceptable approximation result. Yet, the structure is not transparent at the first glance and needs to be documented separately.

Finite Declining Depreciation

Amounts which indicate the reduction in value of fixed assets in profit and loss statements and in cost accounting of a company are called depreciation. Depreciation reflects the obsolescence, market influences and abrasion of fixed assets. The two most widely used methods are linear and declining depreciation.[38]

The method of pure declining depreciation is used in most System Dynamics models. It corresponds to the first-order exponential decay process used in the neoclassical model and reflects the reduction in value sufficiently in most cases, e.g. in the field of macroeconomics.[39] In addition it is quite easy to model by using a first order delay. The depreciation rates are calculated using a fixed percentage on the residual value of an asset (see Formula II). Conducting only this method would never end up in a residual value of 0. Therefore, it is also called infinite depreciation.

Formula II: $d_t = R_t * p$ with: $R_t = R_{t-1} - d_{t-1}$

whereas d_t = depreciation rate in time t
R_t = Residual Value at time t
p = fixed depreciation percentage

---

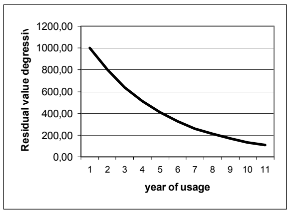
[35] see Brealey, R.A./Myers, S.C. (2000, p. 99)
[36] With T smaller or equal to 5 ordinary methods of algebra are still possible.
[37] see Brown, B.W. (1998, p. 34.1)
[38] These are actually the methods allowed in EStG (Income Tax Act of Germany) § 7. For more information about further methods which can be used for internal cost calculation (e.g. arithmetical-declining, progressive, variable) please refer to Haberstock, L. (1987, pp. 93-107).
[39] see Sterman (2000, p. 441).

When using System Dynamics in the field of planning and budgeting in companies, a more precise calculation of depreciation might become necessary in cases where a result conform to tax laws and commercial laws is essential.[40] These laws regulate that depreciation is only allowed over a fixed and ending period of time. The length cannot be extended or abbreviated. In addition also a maximum fixed percentage for declining depreciation is specified.[41]

After a defined useful economic life the residual value of the original investment has to reach a value of 0. In practice this is mostly realised by switching from declining to linear depreciation. As soon as the linear depreciation rate of the residual value over the residual useful life is higher than the declining depreciation rate the linear depreciation rate is used.[42] An example is provided in Table 4 in addition to the "pure" declining method.

fixed percentage:	20 %
useful economic life:	10 years
investment:	1.000 EUR

	Degressive depriciation		*Switch from degressiv to linear depreciation*			
year	Residual value	depriciation rate	Residual value	degressive depriciation rate	linear depreciation rate	chosen depreciation rate
1	1000,00	200,00	1000	200,00	100,00	200,00
2	800,00	160,00	800,00	160,00	88,89	160,00
3	640,00	128,00	640,00	128,00	80,00	128,00
4	512,00	102,40	512,00	102,40	73,14	102,40
5	409,60	81,92	409,60	81,92	68,27	81,92
6	327,68	65,54	327,68	65,54	65,54	65,54
7	262,14	52,43	262,14	52,43	65,54	65,54
8	209,72	41,94	196,61	39,32	65,54	65,54
9	167,77	33,55	131,07	26,21	65,54	65,54
10	134,22	26,84	65,54	13,11	65,54	65,54
11	107,37	21,47	0,00	0,00	0,00	0,00



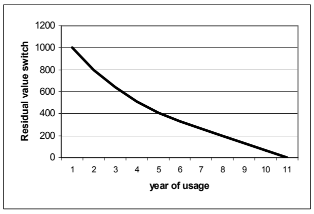


Table 4: Examples for different types of declining depreciation

Implementing this procedure into a System Dynamics model requires calculating all depreciation rates at the time of investment. Otherwise, storing the amount of investment and the useful economic life and calculating residual values and depreciation rates over and over again would slow down the simulation speed. Calculating the depreciation rates at the time of investment requires an iterative process as the rates are always calculated based on the residual value of the asset, not on the amount of investment. In every iteration step, the next declining depreciation rate is compared to a possible switch to linear depreciation. Afterwards the higher value of both is used in order to calculate the residual value. Then, the next iteration step is conducted.

---

[40] In Germany these are EStG (Einkommensteuergesetz; engl.: Income Tax Act), HGB (Handelsgesetzbuch; engl.: Commercial Code).

[41] Since 2001 this fixed percentage is set to 20% per year. Before it was possible to depreciate up to 33% per year (EStG $7(2)2).

[42] EStG § 7a (2)

Value related disinvestment

When disinvesting the number of machines or other objects out of the capital of a company is reduced. This is also reflected in the balance sheet of a company by reducing the monetary value of company assets. It might be sufficient for general analyses to reduce the monetary value by the average value of an asset. That means, the value of disinvestment is the overall value of assets divided by the number of machines/objects.

Yet, this might not be sufficient in the field of planning and budgeting. Herein it might be of high importance which one of the machines/objects is disinvested. The value of older machines is often lower than the value of younger machines due to their loss in value replicated by depreciation. Therefore, it might be important which one of the machines is sold. They generate a different income for the company and have different effects on the balance sheet of the company.

The values of the machines can be calculated based on the depreciation calculation introduced before. It implies the same iteration process and uses the residual value for indicating the possible income.

3.2 Decision Theory

The problem of making decisions is an essential part of our life. Over and over again we face alternatives, choose one of them and go ahead. The effects of those decisions can influence our life sustainable. Formulating and solving problems is the central topic in most scientific disciplines. Therefore, the interdisciplinary research field of decision theory has developed which deals systematically with decision behaviour of individuals and groups. It is, in fact, "concerned with rationality in choice."[43] System Dynamics is used in the field of descriptive as well as prescriptive decision theory.

The aim of descriptive decision theory is to investigate and describe how decisions are made in reality. Consequently, it does not answer questions about how decision should be made, but how decisions actually are made and why. System Dynamics in particular has been used for explaining bounded rationality of decision makers. In fact, "economists who include bounds on rationality in their models have excellent success in describing economic behaviour beyond the coverage of standard theory."[44]

Prescriptive decision theory, in contrast, does not explain reality but gives advices how a decision maker should decide in reality.[45] The essential focus is the choice of one alternative out of different decision/action alternatives.[46] When using System Dynamics as decision support tool those advices should be used and implemented into the models. Mostly decision rules in System Dynamics models choose values of one variable out of a range of possible values, yet they are not concerned with choices, respectively exclusions, of alternatives.

---

[43] Polemarchakis (1991, p. 753).
[44] Conlisk, J. (1996, p. 692).
[45] see Bamberg, G./Coenenberg, A.G. (1996, p. 10).
[46] see Laux, H. (2003, p. 2).

The choice of a certain alternative, e.g. a place of investment or investment into a new type of product, is based on expectations about future conditions and their associated effects. Yet, mostly, those expectations about future conditions are not definite, as different but all realistic expectations might exist. At the point of decision it cannot be foreseen which condition will occur in future. Yet, in some cases the decision maker can give a prediction about the effects of the different conditions on the "goal variable". If it is not possible to predict the probability of occurrence for each condition one speaks about decisions under uncertainty.[47]

The field of decision theory provides approaches of decision making under uncertainty when different alternatives are provided. This paragraph will show different principles about how to deal with decisions under uncertainty that could be implemented into System Dynamics models:

Decision Rule of Maximin

According to the rule of maximin the decision maker should compare alternatives by the worst possible outcome under each alternative, and should choose the one which maximises the utility of the worst outcome. This rule is rational under certain conditions. First, the probability of each circumstance under each decision is unknown. This makes it impossible to calculate expectation of gain. Second, the worst off position chosen by maximin rule is good enough that decision makers are not eager to get more than that. Third, the worst positions under other alternatives are unacceptably bad.[48]

For evaluating an alternative just the most disadvantageous case is taken into consideration. In other words, in this principle the goal is to minimize the risk of loosing. Consequently, it implies a pessimistic decision maker. The objective function can be formulated as:

Formula III: $\min_{c} V_{ac} \rightarrow \underset{a}{Max}!$

whereas R_{ac} = Result of Alternative A_a (a=1,2,..A)
at Condition C_c (c=1,2,…C)[49]

If there is one alternative having the best minimum outcome, the result is unambiguous. Yet, in some cases it might happen that several alternatives result in exactly the same best minimum outcome. In those situations another proceeding decision rule has to be found in order to determine the best alternative.

Based on the assumption of a pessimistic decision maker one could follow the same procedure as in the first step. That means, after excluding those alternatives which are not relevant anymore as their best minimum result is lower than the best of all minimum results, one focuses on the second best minimum result and compares those between the alternatives. This procedure needs to be iterated until either a clear best alternative was found or until the last set of results has been

---

[47] If the decision maker can evaluate the probability of the occurrence of the possible situation the literature speaks about decisions under risk.
[48] http://www.info.human.nagoya-u.ac.jp/~iseda/works/maximin.html (10.01.2003)
[49] see Laux, H. (2003, p. 107), Wöhe, G. (1996, p. 165).

compared. If there is no clear finding after the last step all alternatives, which had the maximum result in the last step, are equal in respect to the chosen criterion.

When implementing the iteration process using auxiliaries and constants the maximum number of iterations has to be modelled explicitly. This structure is quite inflexible concerning changes in ranges. In addition, no matter if there is a definite result after a certain iteration step, all iteration steps need to be calculated as there is no possibility to stop the process when using System Dynamics language.

In order to avoid these disadvantageous the iteration process is transferred into VBS. The main difference to implementing the process using auxiliaries and constants is, first, the iteration process can be stopped as soon as an explicit alternative is found, second just those alternatives which were best at a certain iteration step are considered in the next iteration step.

The rule of Maximin is convenient if the decision environment is judged as being "malicious" instead of being "neutral". It implies an extremely pessimistic attitude of the decision maker. The other extreme is applying the decision rule of Maximax.

Decision Rule of Maximax

According to the rule of maximax the decision maker should compare alternatives by the best possible outcome under each alternative, and should choose the one which maximises the utility of the best outcome. This rule is rational under certain conditions. First, the probability of each circumstance under each decision is unknown. This makes it impossible to calculate expectation of gain. Second, the best position chosen by rule of maximax is just good enough for decision makers who are eager to get the maximum. Third, worse positions under other alternatives are unacceptably bad.[50]

For evaluating an alternative just the most advantageous case is taken into consideration. In other words, in this principle the goal is to maximize the chance of winning and therefore implies an optimistic decision maker. The objective function can be formulated as:

Formula IV: $\max_{c} R_{ac} \rightarrow \underset{a}{Max}!$

whereas R_{ac} = Result of Alternative A_a (a=1,2,..A)
at Condition C_c (c=1,2,…C)[51]

Based on the assumption of an optimistic decision maker one could follow the same procedure as introduced for the rule of Maximin. That means, after excluding those alternatives which are not relevant anymore as their best maximum result is lower than the best of all maximum results, one focuses on the second best maximum results and compares those between the alternatives. This procedure needs to be iterated until either a best alternative is found or until the last set of results has been compared. If there is no definite result after the last step all alternatives, which had the maximum result in the last step, are equal in respect

[50] http://www.info.human.nagoya-u.ac.jp/~iseda/works/maximin.html (10.01.2003)
[51] see Laux (2003, p. 109), Wöhe (1996, p. 166).

to the chosen criterion. Conducting the iteration process using constants and auxiliaries implies the same disadvantageous as for the rule of Maximin. Therefore, a possible solution is implementing it using VBS.

Principle of Hurwicz

The rules of Maximin and Maximax represent borderline cases in the range between pessimistic and optimistic decision makers. Yet, mostly decision makers are not fully pessimistic or optimistic. The principle of Hurwicz tries to overcome this problem by introducing an optimism factor α. The factor is a value between 0 and 1. A value of 0 represents a fully pessimistic decision maker, whereas a value of 1 represents a fully optimistic decision maker. Values in-between give a hint whether the decision maker is rather pessimistic or optimistic. A value of 0.5 represents a decision maker which is risk neutral and therefore neither pessimistic nor optimistic. The chosen factor is multiplied by the maximum value of every alternative and added to the product of the minimum value and 1-α. The alternative with the maximum Hurwicz Weighted Average (HWA) is the preferred one. Consequently, the formulation of the Hurwicz principle is:

Formula V: $\alpha * \max_{c} R_{ac} + (1-\alpha) * \min_{c} R_{ac} \rightarrow \underset{a}{Max}!$

whereas α = optimism factor
R_{ac} = Result of
Alternative A_a (a=1,2,..A) at
Condition C_c (c=1,2,…C)[52]

Conducting the iteration process using constants and auxiliaries implies the same disadvantageous as for the rules of Maximin and Maximax. Therefore it is also suggested transferring it into a VBS.

The Hurwicz Principle already represents a first step to consider more than just the borderline cases of pessimistic and optimistic decision makers. Yet, it only considers two possible results at the same time. Therefore different further possible solutions have been developed in decision theory, but will not be analysed in this project further.

3.3 Optimisation

The issue of combining optimisation and System Dynamics has already been discussed controversial for a long time. The focus has been on two issues where optimisation is particularly useful in the field of System Dynamics, namely model calibration[53] and policy analysis[54]. In both cases, repeated simulation using a specific algorithm (e.g. hill-climbing) is used to adjust selected parameters. Some publications show the practical usefulness and its application.[55]

---

[52] see Laux (2003, p. 110-111), Wöhe (1996, p. 166).
[53] see as examples: Keloharju, R./Wolstenholme, E.F. (1988), Dangerfield, B./Roberts, C. (1999), Kleijnen, J.P.C. (1999).
[54] see as examples: Coyle, G.R. (1985), Gustafson, L./Wiechowski, M. (1986), Keloharju, R./ Wolstenholme, E.F. (1989), Macedo, J. (1989), Graham, A.K./Ariza, C.A. (2003)
[55] see examples in preceding footnotes.

In this paper we refer to another, third, issue in which optimisation can become useful, namely optimisation at a distinct point in simulation time. The usefulness of incorporating optimisation, respectively optimal decision making, into System Dynamics models, has been mentioned just in a few cases.[56] EBERLEIN points out the usefulness in cases "where there are issues requiring allocation or another type of choice. In this setting Linear Programs will often be used to run an optimisation at each time a computation is made."[57]

HOMER, in contrast, warns on including optimisation directly into System Dynamics models. He suggests, in cases where the following two aspects are not fulfilled, System Dynamics and optimization should not be combined in a single model. Rather one model should be build for one purpose.[58] The inclusion of optimisation algorithms works fine as long as:

(1) its time scale is short (hours or days) compared to the SD model's (months or years). Otherwise the parameters of the optimization algorithm, which have to be constant, could change over time. That would turn the static optimisation into a dynamic optimisation process which is not part of this thesis.

(2) one doesn't have to worry about intertemporal issues, where last period's decision could or should have an impact on this period's decision.[59]

In fact, "there are numerous opportunities for system dynamics and hard OR to be used together."[60] The following two possible areas of application – distribution and allocation - show that "simulation and optimisation [can] serve as compliments rather than substitutes".[61] Utilising the possibility of integrating linear programming algorithms into System Dynamics models using Visual Basic scripting overcomes the limitations of System Dynamics, namely insufficient optimization capabilities, but also the limitations of linear programming, namely the static solution.

Distribution problem

When speaking about *the* transportation problem or *the* distribution problem it is referred to the following issue: A certain amount of a homogeneous product s_i (s = source, e.g. factory) is available for distribution at different places $i = 1, 2, \dots m$. At the places $j = 1, 2, \dots n$ a certain amount of the same product d_j (d = destination, e.g. wholesaler, retailer) is needed. The sum of demand $\sum_j^n d_j$ is equal to the sum of products available at the sources $\sum_i^m s_j$. The transportation costs between the places i and j are indicated by c_{ij}. The objective is to specify a transportation plan which guarantees a minimum of transportation costs and a complete fulfilment of the demand. The transported amount is denoted by x_{ij}.

---

[56] see for instance Eberlein, B (1997w); Steinhurst, B. (1997w); Homer, J.B. (1999); Graham, A.K./ Ariza, C.A. (2003, p. 29).
[57] Eberlein, B. (1997w).
[58] see Homer, J. (2000, p. 5).
[59] Result from a correspondence with Jack Homer in Summer 2003.
[60] Homer, J.B. (1999, p. 160).
[61] Brekke, K.A. (2000, pp. 44-45).

The minimized costs are calculated by the costs of transportation c_{ij} multiplied by the amount of transportation x_{ij}:

Formula VI: $C = \sum_{j=1}^{n}\sum_{i=1}^{m} c_{ij} x_{ij} \rightarrow \text{Min!}$

Destinations are served regarding their demand d_j and source s_i have to be depleted completely:

Formula VII: $\sum_{j=1}^{n} x_{ij} = s_i$ and

Formula VIII: $\sum_{i=1}^{m} x_{ij} = d_j$

whereas $x_{ij} \geq 0$ and $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$.[62]

The information above can be summarised in a so called transport tableau as shown in the following table. The fields in the middle of the table (c_{ij}) represent transportation costs per product from source i to destination j. Demand and supply are listed in the last row, respectively in the last column.[63]

The distribution problem can be solved by iteration processes, only. Implementing this process using auxiliaries and constants would become too complex and is not possible to handle efficiently. Therefore, the realisation using the VBS function is suggested. As the optimisation algorithm is conducted each time a computation is made an efficient programming becomes necessary which

- requires little memory space,
- finds the optimum solution quite fast and
- is flexible to changes in ranges.

Sources i	Destinations j				Supply s_i
	1	2		N	
1	c_{11}	c_{12}	...	C_{1n}	S_1
2	c_{21}	c_{22}	...	C_{2n}	S_2
⋮	⋮	⋮		⋮	
m	c_{m1}	c_{m2}	...	c_{mn}	s_m
Demand d_j	D_1	d_2	...	d_n	

Table 5: Distribution problem – transport tableau[64]

In order to meet these requiremente a memory concept is suggested which is based on the ALGOL program published by MÜLLER-MERBACH in 1966.[65] This code does not require saving the whole transportation matrix. Rather, it only needs 2*(m+n) memory places in vectorial form in order to capture the result. The

---

62 see Müller-Merbach, H. (1992, pp. 173-175), Zimmermann, W. (1999, pp. 90-91)
63 Technically, a transport tableau is represented by using multi-dimensional variables.
64 adapted from Zimmermann, W. (1999, p. 91).
65 see Müller-Merbach, H. (1966).

advantage lies in a fast retrieval of the base variables and the usage of less memory space compared to saving the whole matrix.

Some commercial models contain linear programming modules to represent inter-regional transmission.[66] The author of this thesis worked specifically in two projects, one in the field of electricity markets and another in the field of logistics, where inter-regional transportation was part of the model.

Allocation problem

When implementing allocation processes into System Dynamics models, quite often an allocation matrix is specified at the beginning of the simulation and stays fixed throughout the simulation horizon.[67] Yet, a "real" dynamic view should allow an adjusting allocation matrix, based on an optimal solution. The allocation problem is related to the transportation problem. It differs in respect to the amounts of supply s_i and amounts of demand d_j, which are in all cases equal to 1. Moreover the number of suppliers/providers m equals the number of destinations/demanders n. The problem occurs when a pair wise assignment of elements out of two different bulks is conducted based on a specified eligibility as for instance costs or qualification. Examples are the assignment of workers to working places or the marriage problem.[68] The allocation problem can mathematical be formulated as follows:

Formula IX: $C = \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij}$

under the condition that:

Formula X: $\sum_{j=1}^{n} x_{ij} = 1$ and

Formula XI: $\sum_{i=1}^{n} x_{ij} = 1$

whereas $x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$.[69]

A main characteristic of allocation is the problem of degeneration. Every time an allocation has been conducted, two conditions are fulfilled at the same time because of the pair wise assignment. In case the algorithm developed for solving a distribution problem already captures the problem of degeneration it can be used for allocation problems, as well.

A practical case has already been presented by HOMER. In his article about macro- and micro-modelling of field service dynamics a practical case from the repair sector is represented. The assignment of engineers to service jobs is calculated in a linear programming algorithm (Assignment Algorithm). Herein an

---

[66] see Graham, A.K./Ariza, C.A. (2003, p. 29).
[67] see Joglekar, N.R./Ford, D.N. (2002).
[68] see Müller-Merbach, H. (1992, pp. 175 and 276-277).
[69] see Zimmermann, W. (1999, pp. 111)

effective match between the jobs in the queue and the skills of their available engineers is achieved.

4 Prospective

Moving System Dynamics from using it as a mutual learning tool (creating shared mental models with and among stakeholders) to using it as a planning tool, identifying and recommending policy intervention points can require the inclusion of instantaneous processes. If used appropriately and purpose oriented, they can be a basis for valuable models for those trying to extract the maximum they can tell about the underlying system and phenomena of a problem. This article gave first insights into theoretical implications and possible areas of application. In a next step the introduced instantaneous processes will be incorporated into large scale models in order to document the usefulness on practical case studies.

References

Literature

Balci, O. (1998), "Verification, Validation, and Testing", In *The Handbook of Simulation*, J. Banks, Editor, John Wiley & Sons, New York, NY, August, Chapter 10, pp. 335-393.

Balci, O. (2003), "Verification, Validation, and Certification of Modeling and Simulation Applications," In *Proceedings of the 2003 Winter Simulation Conference* (New Orleans, LA, Dec. 7-10). IEEE, Piscataway, NJ, pp. 150-158.

Balzert, H./Loff, J. (1989): Einführung in die Informationsverarbeitung

Bamberg, G./Coenenberg, A.G. (1996): Betriebswirtschaftliche Entscheidungslehre, 9th ed., Munich 1996.

Barlas, Y. (1996): Formal aspects of model validity and validation in system dynamics, in: System Dynamics Review, vol. 12, no. 3, 1996, pp. 1-27.

Brealey, R.A./Myers, S.C. (2000): Principles of Corporate Finance, 6th ed., Boston a. o. 2000.

Brekke, K.A. (2000): Do numerical simulation and optimization results improve management? Experimental Evidence, in: Davidsen, P.I./Ford, D.N./Mashayekhi, A. N.: Sustainability in the Third Millennium - Proceedings of the 18th International Conference of the System Dynamics Society, Bergen 2000, pp. 44-45.

Brockhaus Enzyklopädie (1990), ed. 19, vol. 11, Mannheim 1990.

Conlinsk, J. (1996): Why bounded rationality?, in: Journal of economic literature, vol. XXXIV, 1996, pp. 669-700.

Coyle, G.R. (1985): The use of optimization methods for policy design in a system dynamics model, in: System Dynamics Review, vol. 1, no. 1, 1985, pp. 81-91.

Coyle, G.R./Exelby, D. (2000): The validation of commercial system dynamics models, in: System Dynamics Review, vol. 16, no. 1, 2000, pp. 27-41.

Dangerfield, B./Roberts, C: (1999): Optimisation as a statistical estimation tool: An example in estimating the AIDS treatment-free incubation period distribution, in: System Dynamics Review, vol. 15, no. 3, 1999, pp. 273-291.

Forrester, J.W. (1961): Industrial Dynamics, Portland/OR 1961.

Glenn, J.A. (1984): A dictionary of mathematics, New Jersey 1984.

Graham, A.K. / Ariza, C.A. (2003): Dynamic, hard and strategic questions: using optimization to answer a marketing resource allocation question, in: System Dynamics Review, vol. 19, no. 1, 2003, pp. 27-46.

Gustafsson, L./Wiechowski, M. (1986): Coupling DYNAMO and optimization software, in: System Dynamics Review, vol. 2, no. 1, 1986, pp. 62-66.
Haberstock, L. (1987): Kostenrechnung I – Einführung, Hamburg 1987.
Homer, J.B. (1996): Why we iterate: scientific modelling in theory and practice, in: System Dynamics Review, vol. 12, no. 1, 1996, pp. 1-19.
Homer, J.B. (1999): Macro- and micro-modelling of field service dynamics, in: System Dynamics Review, vol. 15, no. 2, 1999, pp. 139-162.
Homer, J.B. (2000): Notes from the S.D./O.R. Frontier, presentation at the System Dynamics Winter Camp January 2000.
Keloharju, R./Wolstenholme, E.F. (1988): The basic concepts of system dynamics optimization, in: Systems Practice, vol. 1, no. 1, 1988, pp. 65 - 86.
Keloharju, R./Wolstenholme, E.F. (1989): A case study in System Dynamics Optimization, in: Journal of the Operations Research Society, vol. 40, no. 3, 1989, pp. 221-230.
Kerninghan, B.W./Pike, R. (2000): Programmierpraxis. Prinzipien zur effizienten Programmierung, München 2000.
Kleijnen, J.P.C. (1995): Sensitivity analysis and optimization of a system dynamics model: regression analysis and statistical design of experiments, in: System Dynamics Review, vol. 11, no. 4, 1995, pp. 275 – 288.
Krallmann, H. (1994): Systemanalyse im Unternehmen, München 1996.
Laux, H. (2003): Entscheidungstheorie, 5th ed., Berlin a.o. 2003.
Levin, T./Levin, I. (2003): Integrating hybrid modeling with System Dynamics, in: Proceedings of the 21st International Conference of the System Dynamics Society, ed. by Eberlein, R.L./ Diker, V.G. / Langer, R.S. / Rowe, J. I., New York 2003, pp. 83-84.
Liedl, R./Reich, E./Targonski,G. (1985): Iteration Theory and its functional equations. Proceedings, Schloss Hofen 1984, Berlin/Heidelberg 1985.
Love, G. (2001): Hybride Modelling: Embedding a LP Solver in a SD model, in: Proceedings of the 19th International Conference of the System Dynamics Society, ed. by Hines, J.H./ Diker, V.G. / Langer, R.S. / Rowe, J. I., Atlanta 2001, pp. 99-100.
Macedo, J. (1989): A reference approach for policy optimization, in: System Dynamics Review, vol. 5, no. 2, 1989, pp. 148-175.
Mertens, P. (1982): Simulation, 2nd ed., Stuttgart 1982.
Müller-Merbach, H. (1966): Die Lösung des Transportproblems auf Rechenautomaten – ein ALGOL Programm, in: Elektronische Datenverarbeitung, vol. 8, no. 2, 1966, pp. 49 – 56.
Müller-Merbach, H. (1992): Operations Research. Methoden und Modelle der Optimalplanung, 3rd ed., München 1992.
Osgood, N. / Kaufmann, G. (2003): A hybrid model architecture for strategic renewable resource planning, in: Proceedings of the 21st International Conference of the System Dynamics Society, ed. by Eberlein, R.L./ Diker, V.G. / Langer, R.S. / Rowe, J. I., New York 2003, pp. 100-101.
Perridon, L./Steiner, M. (1997): Finanzwirtschaft der Unternehmung, 9th ed., Vahlen 1997.
Polemarchakis, H.M. (1991): Decision Theory, in: The New Palgrave. A dictionary of economics, vol. 1 A to D, ed. by Eatwell, J./Milgate, M./Newman, P., London, pp. 753-756.
Rakitin, S.R. (1997): Software Verification and Validation. A Practitioner's Guide, Norwood 1997.
Randers, J. (1980): Guidelines for Model Conceptualization, in: Elements of the System Dynamics Method, ed. by Randers, J., Cambridge (Mass.) 1980.

Sterman, J.D. (2000): Business Dynamics: Systems Thinking and Modelling for a complex world, Boston a.o. 2000.
Wöhe, G. (1996): Einführung in die allgemeine Betriebswirtschaftslehre, 19th ed., München 1996.
Wolstenholm, E.F. (1990): System Enquiry: A System Dynamics Approach, 1990.
Zimmermann, P.A. (1987): Test-Techniken: Methodisches Testen von Systemen und Programmen, Vaterstetten 1987.
Zimmermann, W. (1999): Operations research: Quantitative Methoden zur Entscheidungsvorbereitung, 9th ed., München/Wien/Oldenburg 1999.

Laws and Guidelines
VDI Guideline 3633 Part 1 (1993): Simulation of systems in materials handling, logistics and production. VDI-Gesellschaft Fördertechnik Materialfluss Logistik (VDI-FML), Düsseldorf 1993.
WWW-Resources
http://www.info.human.nagoya-u.ac.jp/~iseda/works/maximin.html (10.01.2003)
http://www.lsbu.ac.uk/php-cgiwrap/immage/PGprojects.php3 (04.09.2003)
Eberlein, R. (1997w): http://www.optimlator.com/master/97B/1421.htm (17.05.2002).
High Performance Systems: www.hps-inc.com
Powersim Software: www.powersim.com
Steinhurst, B. (1997w): http://www.optimlator.com/master/97B/1426.htm (17.05.2002).
Ventana Systems: **www.ventanasystemsinc.com**