© IEEE – Original final publication: Al-Begain, K., Al-Dabass, D., Orsoni, A., Cant, R., Zobel, R. (eds.) : 2013 8th EUROSIM Congress on Modelling and Simulation; EUROSIM 2013,Cardiff, Wales, United Kingdom, September 10-13, 2013; pages 336–344
https://ieeexplore.ieee.org/document/7004966

Assessing Risk in Discrete Event Simulation by Generalized Deviation

Arne Koors
Department of Informatics
University of Hamburg
Hamburg, Germany
e-mail: koors@informatik.uni-hamburg.de

***Abstract*—This paper introduces a generalized deviation concept, inspired by quantitative finance. Standard risk metrics like volatility or downside risk are deconstructed into five general sub-functions for reference, selection, penalization, normalization and re-dimensioning. The advantage of this approach is its flexibility, allowing modeling a wide range of risk perceptions in numerous application fields of discrete event simulation. Several further risk types like upside risk, outside risk, transition risk, critical state risk or countermovement risk are describable and embeddable as special cases of generalized deviation. These risk types are presented with respect to motivation, specification of relevant generalized deviation components, description of application classes, an application example and a graphical illustration. In particular, various options for determining reference states, reference selection and penalty functions are discussed. Implementation features of the generalized deviation metric in the discrete event simulation framework DESMO-J are outlined. Moreover, possible structural extensions as well as additionally implementable risk types are delineated, indicating further application potential and the flexible scope of this approach. It is proposed to complement descriptive standard statistics in discrete event simulation domains by additionally employing risk measurement in terms of generalized deviation as explicated here, to facilitate assessment of undesired simulation dynamics in various application fields.**

Keywords-risk; risk metrics; generalized deviation; discrete event simulation

I. INTRODUCTION

Discrete event simulation is a methodology for modeling dynamic processes in real or imaginary systems, executing experiments with the aim of gaining insights which can be re-transferred to the investigated original system [1]. In order to actually gain insight and assess a modeled system, system output variables of interest are chosen by the modeler, and selected statistics on these state variables are collected during experiment run time. They are then shown in the simulation report at the end of an experiment.

Typically, descriptive standard statistics are calculated, like mean, variance, standard deviation, minimum, maximum or simply the number of observations. Further statistics include median, quantiles, modes, skewness or kurtosis; more advanced concepts deal with histograms, regression analysis, correlation, confidence intervals or time-weighted versions of the statistics mentioned above.

Each of the aforementioned statistics describes one specific aspect of the observations made in a rather objective way, leaving it to the experimenter to assess welcome or undesired outcomes in the aftermath.

Discrete event simulation often is employed to evaluate design alternatives of new systems or optimize already existing systems. For this purpose, it is essential to assess and rate the system's state variables dynamics concerning welcome or undesired behavior. The conventional statistics mentioned before do not contain intrinsic valuation standards, therefore there is no choice but to base valuation on already aggregated figures, with no way to weight elementary observations according to the specific simulation study's aims and needs.

E.g. with given mean and variance of cost, variance of cost below the mean normally is welcome, whereas variance of cost above the mean is undesired. However, use of already aggregated statistics (here: variance) restricts appropriate downstream valuation: In conventional discrete event statistics, there are no metrics offered to separate variance in welcome and undesired sections.

In principle it is feasible to manually implement customized performance metrics for any specific application case. Nevertheless it seems preferable to avoid repeated implementation effort by providing a general framework for dynamics assessment, to generically cover a wide range of discrete event simulation application fields.

In this paper, a generic class of additional simulation performance metrics called *generalized deviation* is proposed, enabling the modeler to accurately control aggregation of elementary observations according to objects and valuation standards of a study. Once implemented in existing simulation software (like described here for the framework DESMO-J [2], [3]), this class of metrics enables flexible assessment of simulation dynamics by composing functional building blocks – which are predefined or may be self-developed – without further technical implementation effort.

In [4] it was proposed to transfer financial risk metrics to discrete event simulation, namely Semi-Variance, Value at Risk, Expected Shortfall (also called Conditional Value at Risk) and Drawdown. A forthcoming paper reports on implementation and visualization of these metrics in the discrete event simulation framework DESMO-J [5]. It has been found that financial risk metrics are inspiring for assessment of dynamics in discrete event simulation, but need further generalization for wider applicability. The generalized deviation metric introduced here draws its intellectual stimulus from the quantitative finance notion of measuring uncertainty of returns by standard deviation and below-expected returns by semi-deviation. This contribution reports on extension and formal generalization of these two metrics and showcases how typical application fields of discrete event simulation may benefit from the generalized deviation metric to assess undesired model dynamics in a case-specific manner, far beyond the original financial use cases.

This paper is structured as follows: Section II defines the proposed generalized deviation metric formally and illustrates it by the example of standard deviation. Section III generalizes classical financial risk metrics like volatility and semi-deviation and embeds them into the more flexible generalized deviation concept. Furthermore, several advanced risk notions in discrete event simulation application fields are delineated, illustrating the application potential and flexible scope of this approach. In particular, various options for determining reference states, reference selection and penalty functions are discussed. Section IV concerns interpretation of generalized deviation. Section V outlines the software implementation in the discrete event simulation framework DESMO-J. Section VI concludes the paper and gives an overview of further implementable risk types and possible structural extensions.

II. Generalized Deviation Metric

Quantitative Finance deals with computer-assisted analysis of asset prices and supports investment decisions in financial markets. One of its most significant assessment categories relates to the term *risk*. A *risk metric* is a concept to quantify financial risk, in order to compare different portfolios or trading strategies with respect to the risk taken.

This section generalizes one approach to the quantitative finance concept of risk and applies it to discrete event simulation, *regarding risk as observation of undesired state dynamics of a modeled system*. For this purpose, we introduce a generalized deviation metric, where risk can be measured and expressed by definition of five sub-functions for reference, selection, penalization, normalization and re-dimensioning.

A. Definintion

Be $\mathbb{T}$ the data type of simulation time instants with a defined operator $\leq$ for ordering simulation time instants. Be v a state variable with possible states $s \in \mathbb{S}$ and be $o_i = (t_i, s_i) \in \mathbb{T} \times \mathbb{S}$ an observation of v with an observed state s_i at simulation time instant t_i. Be $\mathbb{O} = \mathbb{T} \times \mathbb{S}$ the set of all possible observations and $O = \bigcup_i o_i, O \subseteq \mathbb{O} \Leftrightarrow O \in \mathcal{P}(\mathbb{O})$ the set of all observations of v actually recorded during a simulation run. Assuming observations o_i are stored successively during simulation implies that they are ordered in simulation time, so that $\forall i: t_i \leq t_{i+1}, 1 \leq i < |O|, i \in \mathbb{N}$. ("$\mathbb{N}$" here represents the set of natural numbers, whereas all other set letters – even "$\mathbb{R}$" – in this paper represent freely selectable sets, unless otherwise stated.)

Then, a *Generalized Deviation* metric *GD* of observations O may be defined as shown in (1).

Here, $penalty: \mathbb{N} \times \mathcal{P}(\mathbb{O}) \times \mathbb{R} \rightarrow \mathbb{X}$ is a function that assesses risk towards each single *i*th observation in O with regard to a corresponding reference, returned by the function $select: \mathbb{N} \times \mathcal{P}(\mathbb{O}) \times \mathcal{P}(\mathbb{R}) \rightarrow \mathbb{R}$. The *select* function chooses one reference out of a set of (observed or computed) alternative references, generated by the function $reference: \mathbb{N} \times \mathcal{P}(\mathbb{O}) \rightarrow \mathcal{P}(\mathbb{R})$. The penalties of all observations in O are added and afterwards normalized by a function $normalize: \mathbb{X} \times \mathcal{P}(\mathbb{O}) \rightarrow \mathbb{Y}$. Finally, potential changes in dimension induced by the penalty function are compensated by a function $redimension: \mathbb{Y} \times \mathcal{P}(\mathbb{O}) \rightarrow \mathbb{Z}$.

B. Example

As an example, we can express the quantitative finance risk metric *Volatility* (which simply is the standard deviation) of a set of observations O

$$SD(O) = \sqrt{\frac{1}{|O|-1}\sum_{i=1}^{|O|}(s_i - \bar{s})^2}$$

by $GD(O)$ as defined above, with the sub-functions

$$reference(i, O) = \left\{\frac{1}{|O|}\sum_{j=1}^{|O|} s_j\right\},$$

$$select(i, O, \{r\}) = r,$$

$$penalty(i, O, r) = (s_i - r)^2,$$

$$normalize(x, O) = \frac{x}{|O|-1},$$

$$redimension(y, O) = \sqrt{y},$$

and $\mathbb{R}, \mathbb{S}, \mathbb{X}, \mathbb{Y}, \mathbb{Z}$ all equal to the set of real numbers.

In this example, the *reference* function ignores its first parameter *i* and simply computes the mean $\bar{s}$ of all observed states s_j in the observation set *O*, given as second parameter. Thus, the reference state constantly is the mean of states, independent of the current observation under examination.

Furthermore, the reference function generates a set

$$GD(O) = redimension\left(normalize\left(\sum_{i=1}^{|O|} penalty\Big(i, O, select\big(i, O, reference(i, O)\big)\Big), O\right), O\right) \tag{1}$$

containing only one single reference per observation without other alternative references to choose from. Thus, the *select* function merely passes on the (mean) reference to the downstream *penalty* function.

The *penalty* function utilizes its first two parameters *i* and O to determine the *i*th observation o_i in the observation set O; note that observations o_i can be ordered by time instant t_i, as postulated above. Then, the second component of observation o_i (i.e. the actually observed state s_i) is accessed and the selected reference given as third function parameter $r(= \bar{s})$ is subtracted from s_i's value. The difference is squared. This expresses a nonlinear, over-proportional risk attribution: deviations from the reference mean state are penalized quadratically.

The *normalize* function divides the sum of penalties by the number of observations minus one, because an estimator for the unbiased sample variance of quasi "empirical" simulation observations is calculated. The second function parameter containing the set of observations O is only used to determine the number of total observations $|O|$.

By penalizing observed states in computing the squared difference to the mean of states, the dimension of the metric was squared as well. This finally is compensated by the *redimension* function, which simply calculates the square root of the normalized squared difference sum, given as first parameter. The second parameter containing the set of observations O is disregarded in the context of the standard deviation. To keep options open, the *redimension* function may be defined rather freely. Nevertheless, it can be expected that it will regularly contain an inverse function of a component of the penalty function.

In this way the volatility or standard deviation can be expressed as one special instantiation of the generalized deviation metric introduced above.

III. Risk Types and Application Examples

In this section, we will elaborate on different risk types that can be expressed by instantiation of the five sub-functions of generalized deviation (abbreviated in the following as "GD"). Each risk type is covered regarding motivation, specification of the relevant GD sub-functions, a description of application classes, an example and a graphical illustration.

A. *Two-sided Deviation from a Representative State*

In quantitative finance, the volatility resp. standard deviation is utilized when regarding risk as uncertainty of return around a central mean return [6]. In this context, return is no genuine observation, but derived from change of asset prices within a certain time span.

In discrete event simulation, observations of freely chosen state variables can be recorded into time series. The arithmetic mean is not necessarily an appropriate characterization of observed state values, because it is sensitive to statistical outliers. In many cases, the median or the (most frequent) mode state may be more representative choices. In these cases, the reference function of generalized deviation can simply be set to

$$reference(i, O) = \{median(O)\} \text{ or}$$

$$reference(i, O) = modes(O).$$

Whereas there is only one median of an observation recording, there may exist more than one mode, because two or more mode states may be observed with the same maximum frequency. Therefore a set containing more than one possible reference may result from the reference function, and it is task of the downstream select function to determine which of the reference alternatives is processed further by the penalty function. An obvious option is to choose the reference alternative that is nearest to the current state under examination s_i by

$$select(i, O, R) = nearest(i, O, R) :=$$

$$r \in R \text{ so that } \forall r_j \in R{:}\ |r_j - s_i| \geq |r - s_i|, r_j \neq r.$$

Another option is setting the reference state to an externally specified value, which could be a desired target state considered "best" for the state variable by the modeler. In this case, all deviations from the desired target state can be interpreted as risk and will be revealed in the simulation report.

Regarding standard deviation, deviation from the mean reference state is penalized quadratically. In general, risk perception in arbitrary application domains may differ from this view. Accordingly, the penalty function of generalized deviation can be customized to e.g.

$$\text{exponential: } penalty(i, O, r) = e^{|s_i - r|} - 1,$$

$$\text{linear: } penalty(i, O, r) = |s_i - r|,$$

$$\text{square-rooted: } penalty(i, O, r) = \sqrt{|s_i - r|},$$

$$\text{logarithmical: } penalty(i, O, r) = ln(|s_i - r| + 1)$$

or any other function that appears adequate in the application domain. In the majority of cases, it seems advisable that the penalty function is monotonically increasing from the point of origin and symmetrical w.r.t. the y-axis. Fig. 1 depicts the graphs of the penalty functions proposed above; $s_i - r$ is

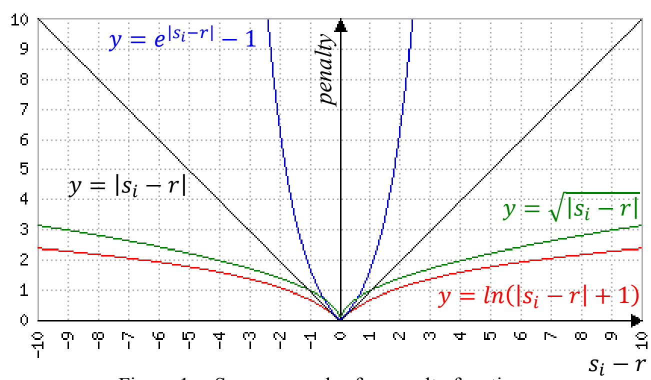

Figure 1. Some examples for penalty functions.

assigned to the x-axis.

The redimension function has to be adjusted according to the penalty function, to ensure that generalized deviation and observed states are of the same dimension. Ignoring this step would mean that observed states and the GD value do not correlate reasonably. For the penalty functions given as examples above, the redimension functions should be

$$\text{logarithmical: } redimension(y, O) = ln(y),$$

$$\text{linear: } redimension(y, O) = y,$$

$$\text{quadratic: } redimension(y, O) = y^2, \text{ resp.}$$

$$\text{exponential: } redimension(y, O) = e^y.$$

A typical application class where GD measures risk as two-sided deviation from a representative reference state are systems in equilibrium or in default states, when deviation in any direction is regarded as risk.

Consider the simulation of a warehouse that has been built as a buffer between producers and consumers. Here, undesired dynamics (risk) occurs, if the warehouse filling level tends towards its maximum capacity. When reaching this point, producers could not store their goods immediately and would have to wait for consumers until unloading is possible again, entailing additional cost for waiting time and non-availability of trucks at other locations. On the other hand, a filling level too low would threaten that goods run out of stock and a sudden peak of consumer demand might leave part of the consumers un-served, resulting in loss of customers or even compensatory claims resp. penalty payments. Here risk arises as deviation in both directions, regarded from a central, "safe" system state.

Fig. 2 illustrates warehouse utilization dynamics and shows risk extent, in this case the average absolute linear deviation of approx. 20% from the mode warehouse utilization level of 44%. The chart is part of the simulation output of our open source discrete event simulation framework DESMO-J, developed at the University of Hamburg. The set GD parameters are displayed at the

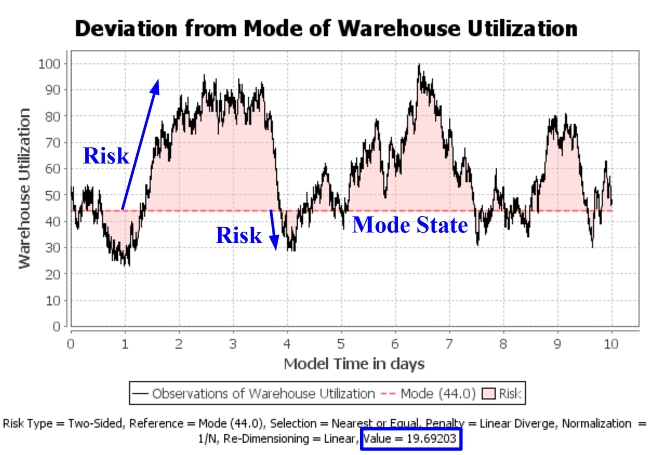

Figure 2. Risk as two-sided deviation from the mode state.

bottom of the chart. They are included in the textual simulation report as well, as shown in section V. Annotations in blue color have been added manually.

Incidentally, it should be noted that risk and *cost* are independent of each other: varying risk may be measured without a change of cost in the system and cost may alter without affecting risk. In the warehouse example, risk expresses the probability of undesired extra costs; but as long as risk is not extreme, no additional costs actually arise in daily warehouse operation, although risk permanently changes corresponding to warehouse filling levels. On the other hand, streamlining business processes in warehouse operation may reduce (e.g. overhead) cost, but risk may stay unalteredly at the same levels as before.

B. *One-sided Deviation from a Reference State*

In quantitative finance, it has been argued that negative deviations from the expected (mean) return indeed pose a risk, whereas positive return deviations are welcome and should not be incorporated into risk metrics. As a consequence, the concept of *Downside Risk* was developed [7], [8], [9], [10], where positive deviations from the mean are simply replaced by the mean, when computing risk. The most prominent downside risk metric is *Lower Semi-Deviation*, which again can easily be expressed as a special case of generalized deviation: To obtain this metric, the same sub-functions as for the standard deviation given in sub-section II.B are applicable, except for

$$select(i, O, \{r\}) = \begin{cases} r, & s_i < r \\ null, & s_i \geq r \end{cases}$$

$$penalty(i, O, r) = \begin{cases} (s_i - r)^2, & r \neq null \\ 0, & r = null \end{cases}$$

where $null \in \mathbb{R}$ is a special value in the set of references $\mathbb{R}$, indicating that none of the references generated by the reference function was selected. The small modification above leads to only incorporating states s_i below the mean $r (= \bar{s})$ in penalty computation. It is obvious that standard deviation and lower semi-deviation are closely related, since only two of the five sub-functions of generalized deviation are modified slightly. The structural difference of the two risk concepts solely originates from different selection (resp. ignorance) of reference states.

The concept of asymmetric risk assessment is applicable well beyond finance and is easily implementable by a case differentiation in the GD select function, as shown above. Nevertheless, the diverse application fields of discrete event simulation will not generally prefer lower observation states to higher states. Specific demand for complementary *Upside Risk* metrics can easily be satisfied by flexibly exchanging terms for $s_i < r$ and $s_i \geq r$ in the select function, now penalizing positive deviations in consequence.

In the context of downside and upside risk, mean, median and mode have useful applications as reference states. However, due to asymmetric risk perception, one state of all observations will be most distant from undesirable state regions. This will be an extremum state, the minimum or maximum of all observed states, and thus setting the reference function to

$$reference(i, O) = \{min(O)\} \text{ or}$$

$$reference(i, O) = \{max(O)\}$$

supports further use cases.

A typical application class where GD measures risk as one-sided deviation from a reference state are systems with state variables that have a preferred extremum boundary state (e.g. an empty waiting queue), where one-sided departure from this (extremum) state is regarded as risk.

For example, consider a logistics simulation of operating a truck fleet. Overall utilization near the maximal value of 100% load capacity is highly desirable for ecological and economic reasons. GD may measure all downside deviation from this level as unwelcome risk; fig. 3 displays an application scenario where the maximum of observations was chosen as reference state and risk is assessed as downside deviation from the maximum.

C. *Deviation from a Reference State Channel*

In the preceding sub-section III.A, two-sided deviation from one constant representative state was discussed. In doing so, the reference state is considered riskless, because $|s_i - r|$ is zero for $s_i = r$ and a penalty function typically will yield 0 for deviation 0: with no deviation, there is no risk. Generally, in discrete event systems there may exist more than one riskless state, for example a whole interval of states regarded as riskless. In this case, two reference states act as the boundaries of a safe interval resp. state channel. Deviations outside of the channel are penalized as *Outside Risk*, according to the distance from the nearest channel boundary. This scenario is expressible in the generalized deviation framework by setting

$$select\left(i, O, \{r_{lower}, r_{upper}\}\right) = \begin{cases} r_{upper}, & s_i > r_{upper} \\ r_{lower}, & s_i < r_{lower} \\ null, & \text{else} \end{cases}$$

with a reference function that statically generates the set of the two interval boundaries r_{lower} and r_{upper}. The select function chooses the upper (resp. lower) channel boundary for all states above (resp. below) the corresponding boundary and returns *null* for states within the channel boundaries. Consequently, the penalty function has to be adjusted to ignore *null* reference values, as shown in sub-section III.B.

Further, the normalization function should be adapted to consider only valid non-*null* selections: The set of uncritical observations within the channel, yielding to *null*-selections, should not distort risk computation for the remaining critical observations outside of the safe channel. This can be achieved by setting

$$normalize(x, O) = \frac{x}{|O| - |null\ selections\ in\ O|}$$

A typical application class where GD measures outside risk as deviation from a reference state channel are systems in a stationary phase, where state fluctuation within a certain state interval is tolerable, but deviation beyond given state boundaries is regarded as risk.

A different example in manufacturing simulation is given below. Here, perpetual workload of 100% in e.g. final assembly may decrease workers' output quality. Thus short breaks at the workplace between pieces of work might be advisable both for health of workers and reduction of rework cost. On the other hand, insufficient capacity utilization would be unacceptable because of high fixed personnel costs. Here, risk can be attributed to deviation of workload into upward as well as downward direction, with respect to a channel of preferred variable states. Fig. 4 displays an appropriate scenario where GD measures risk as deviation outside of a preferred resp. safe reference state channel.

D. *Transition between Reference Phases*

In general, there may be more than one interval of safe states for a modeled system. For example, a system may have two or more stationary phases, and risk emerges from transition between these phases, because during phase transition the system is in an unstable and thus vulnerable

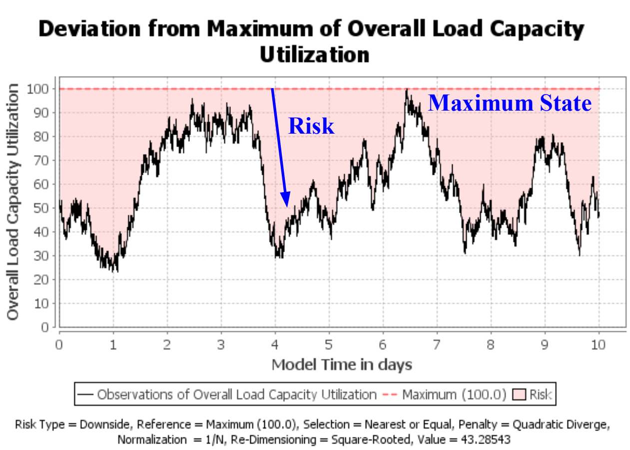

Figure 3. Risk as downside deviation from the maximum state.

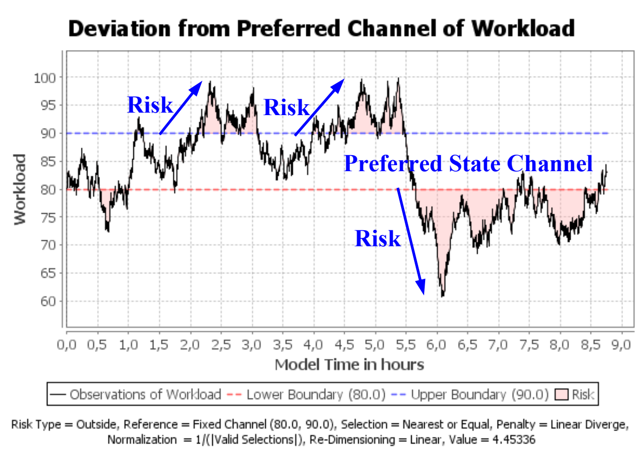

Figure 4. Risk as deviation outside of a preferred state channel.

state.

Transition Risk can be modeled by

$$select(i, O, R) =$$

$$\begin{cases} r = nearest(i, O, R), s_i > r \wedge type(r) = UpperBoundary \\ r = nearest(i, O, R), s_i < r \wedge type(r) = LowerBoundary \\ \qquad\qquad null, \text{else} \end{cases}$$

with a reference function that statically generates upper and lower boundaries of reference phases, and an auxiliary function $type: \mathbb{R} \rightarrow \{UpperBoundary, LowerBoundary\}$ that determines whether a reference is an upper or lower boundary of a phase. Then, the select function chooses the nearest boundary of neighboring phases, if the currently observed state is between two phases; resp. *null* if the observed state is within a phase. Penalty and normalization functions should be adjusted as advised in sub-section III.C.

A typical application class where GD measures transition risk are bistable or multistable systems, where state fluctuations within stable phases are tolerable, but state dynamics outside of these safe phases are regarded as risk.

Consider the simulation of social agents in a political two party system. There may be two smaller core groups that always remain loyal to their respective political parties, and a third large group of swing voters that are the decisive factor for election results. A new government will act more effectively the higher the supporting percentage of population is. Conversely, it would be a sign of inner disunity, if almost half of voters balloted against the future government. Then, legitimation of the government may be questioned and social tensions might arise. In this system, poll ratings that clearly favor one of the two parties may be interpreted as evidence for desired stable political and social conditions (as long as minor party proponents are not discriminated), whereas transitions between any of the two stable phases may indicate dissatisfaction and inner disunity, in other words social risk. Fig. 5 displays an application scenario where GD measures risk in terms of transitions between two phases of stability.

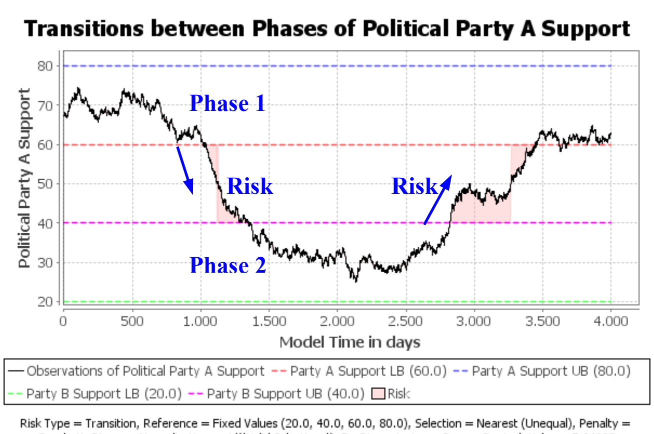

Figure 5. Risk caused by transition between stable reference phases.

E. Approach towards a Reference State

The risk types discussed so far have in common that reference states r are linked to the set of observations O. Typically, prominent states like median, mode, extremum or boundary states are chosen as reference, which actually have been observed (or should have been observed, in the case of externally specified most desirable reference states). These states resp. state intervals are considered riskless, and penalty of an observed state s_i is assessed according to s_i's distance from a riskless reference state r, utilizing the term $|s_i - r|$. The penalty function typically increases monotonically, since a higher distance from riskless states is associated with more risk.

This notion of quantitative finance is not suitable for all of the diverse discrete event simulation application fields. Frequently, risk will not be defined by deviation from riskless states but by approaching towards risky states. Sometimes riskless states may be irrelevant or unspecified, and often simulation models will be concerned with whether certain critical states are reached, then damaging or irrecoverably destroying the system under examination.

This "classic", sometimes more natural *Critical State Risk* can be expressed by setting risky instead of riskless states as reference states and employing a monotonically decreasing penalty function, e.g.

$$\text{hyperbolic: } penalty(i, O, r) = \begin{cases} \frac{1}{|s_i - r|}, & s_i \neq r \\ +\infty, & s_i = r \end{cases},$$

where penalty is infinite, if a critical reference state is reached and the system is destroyed. If critical states only represent system damage, the penalty function may be adjusted to approximate a given high positive figure for $s_i = r$. Thus, repeated damage will be reflected in higher overall risk. If the actual distance from risky reference states r is irrelevant and solely critical events are under consideration, penalty for $s_i \neq r$ may be set to 0 and penalty for $s_i = r$ to e.g. 1. Thus, the number of critical incidents will be counted.

A typical application class where GD measures risk as approach towards a critical reference state are systems in operation, when state fluctuations are tolerable in principle, but approaches towards certain states of (partial) damage or system destruction are regarded as risk.

When e.g. simulating scheduling and machine utilization strategies in production, a number of organizational and technical boundary conditions have to be satisfied. For example, machines may have a maximum operating temperature recommended by the manufacturer and a somewhat higher temperature at which damage actually occurs. Any exceeding of the manufacturer's maximum temperature threshold and approach towards damaging machine temperature is a risky operation policy, be it due to overutilization without pause, lack of maintenance or insufficient cooling. Fig. 6 displays an application scenario where GD measures risk as approaching a critical reference state. Here, risk is only attributed to those states higher than a threshold that separates critical from uncritical states. – In this example, operation is interrupted by a forced break after reaching a critical state and resumed some time later.

F. *Deviation from Previous Extremes*

In all scenarios discussed so far, generated reference states remain constant throughout the whole simulation experiment. This need not necessarily be so. In quantitative finance, the term *Drawdown Risk* expresses the possibility of undesirably losing asset value that has been gained before. In other words, the reference state that penalty refers to is the maximum of all previously observed states s_i:

$$reference(i, O) = \{max(\{s_j | 1 \leq j \leq i\})\}.$$

In consequence, states s_i that were riskless or had a low risk in the past may be considered more and more risky in the course of simulation time, as a result of rising maximum levels.

Here again, the diverse application fields of discrete event simulation will not generally prefer rising observation states to falling states. Complementary *Runup Risk* can easily be defined by substituting the max term in the reference function above by the min function, setting a sequence of falling minimums as reference for penalization.

A typical application class where GD measures risk as deviation from previous extremes are systems with state variables that consistently grow or decline. Here, current extremum states only have temporary relevance and risk can be regarded as countermovement of variable states against the main direction of development.

E.g. in simulation of an epidemic in a population of agents, all decreases in infection rate are favourable, and every new minimum min_t of infection rate will be welcome. When infection rate declines further and new minimums min_{t+n} are reached, increases of infection rate back to then-past minimum levels min_t now will be undesired. Moreover, any countermovement against the prevailing falling trend will be considered as risk of a new epidemic outbreak. Fig. 7 displays an application scenario where GD measures risk as countermovement against a sequence of falling previous minimum states.

IV. Interpretation

The sub-sections III.A to III.F have discussed a number of generalized deviation instantiations for different risk notions and application classes. However the constitutive sub-function components are defined in particular, computation of GD finally results in a scalar value.

In the first place, this GD value quantifies the normalized (e.g. average) intrinsic risk of observed state development in the simulated system. Correspondingly, the reference function should be set to generate the boundary of (un)risky states. The select function should separate riskless from risky states and choose between risk reference alternatives. The penalty and redimenson functions should express the nature of risk perception. In such a model-adjusted setting, the obtained GD value yields a concrete meaning, which is interpretable as quantification of risk in the context of the modeled system.

Further interpretation of GD corresponds to handling of standard deviation values SD obtained from conventional statistics in simulation reports. First, GD may be used to get an impression on how distant risky states are distributed from their reference states (in terms of absolute distance). Second, GD may be set into relation to the selected references, analog to calculating the ratio $SD/\bar{x}$ in classical descriptive statistics, to assess relative dispersion. While this is easily feasible for cases of constant reference selection, relating to multiple or non-static references as covered in sub-sections III.C, III.D and III.F will involve additional computation. Third, GD may be used as a means of relative comparison between simulation experiments with different random number seeds or parameter settings. Fourth, GD might be utilized to determine whether a particular observed state s_i may be classified as a remarkable outlier, in terms of high risk – e.g. if s_i is beyond a distance of three general deviations from its reference state. It should be noted that this usage scenario has to be implemented cautiously, because risk commonly will not be normally distributed, and therefore there is no general law of how much risk can be expected per GD unit of distance from reference states.

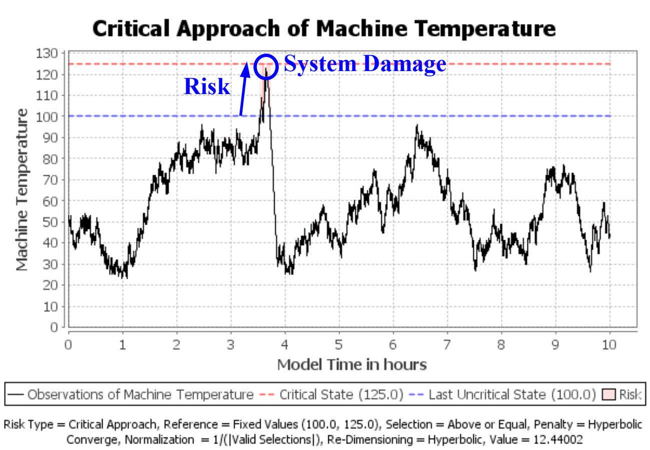

Figure 6. Risk as approach towards a critical reference state.

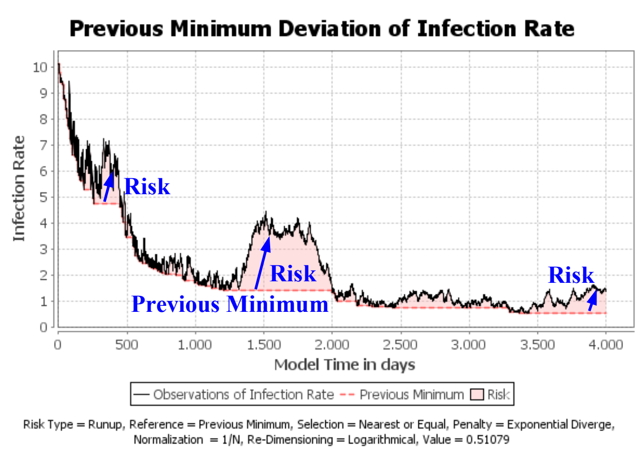

Figure 7. Risk as deviation from previous extremum states.

V. IMPLEMENTATION

The generalized deviation metric has been implemented in the open source discrete event simulation framework DESMO-J [2], [3], developed at the University of Hamburg. The GD sub-framework consists of 1 package, 6 sub-packages and 58 Java classes.

`Recording` objects maintain `Lists` of `Observation` objects o_i. Observations can be recorded during simulation by invoking the `recording.update()` method. Observations store their observed time instant t_i and state s_i (cf. sub-section II.A). A `Recording` object corresponds to the (chronologically ordered) set O of all observations of a state variable v in a simulation run.

The central class `GeneralizedDeviation` has a constructor that takes a model reference, a title string, a `Recording` object and a risk type as parameters. Further, the constructor expects five parameter objects of types `Reference`, `Selection`, `Penalty`, `Normalization` and `Redimensioning`. These types are predefined Java interfaces; `Reference` is an abstract class.

Objects of these five data types have to implement a `getName()` method (for textual output in the simulation report, fig. 8) and a computation method, as defined in sub-section II.A. When the `getValue()` method of a `GeneralizedDeviation` object is called, all observations o_i of its recording object are processed in a loop, by calling the computation methods of the involved five GD component objects, in the order specified by (1) in section II.A:

```
// ...
for (int i = 0; i < _recording.length(); i++) {
    alternatives = _reference.generateReferences(i, _recording);
    reference = _selection.select(i, _recording, alternatives);
    if (reference != null)
        cumulatedPenalties += _penalty.penalty(i, _recording,
                reference);
}
value = _redimensioning.redimension(_normalization.normalize(
        cumulatedPenalties, _recording,
        _reference.getReferenceData()), _recording);
// ...
```

42 building block classes for the five GD component types have been predefined and implemented, enabling the modeler to easily compose case-specific risk metrics according to his/her simulation study's aims and valuation standards.

Particularly common combinations of building blocks are encapsulated in 8 additional classes, e.g.

```
public StandardDeviation(Model ownerModel,
        NumberRecording<T> recording,
        boolean showInReport, boolean showInTrace) {

    super(ownerModel, "Standard Deviation", recording,
            "Uncertainty", new Mean<T>(), new NearestOrEqual<T>(),
            new QuadraticDiverge<T>(), new Corrected<T>(),
            new SquareRooted<T>(), showInReport, showInTrace);
}
```

– compare this code to the corresponding example in sub-section II.B.

The necessity to pass objects like `new Mean<T>()`, which implement a computation method to e.g. calculate the mean of a recording's observed states, originates from the inability of Java to handle functions as parameters. In a Scala or C# implementation, one would pass functions directly instead.

New generalized deviation objects can be built ad hoc in one statement, e.g.

```
previousMinimumDeviation =
        new GeneralizedDeviation<Double>(
            this, "Previous Minimum Deviation",
            infectionRateRecording, "Runup",
            new PreviousMinimum<Double>(),
            new NearestOrEqual<Double>(),
            new ExponentialDiverge<Double>(),
            new Uncorrected<Double>(),
            new Logarithmical<Double>(), true, false);
```

– this code generated fig. 7 and the first line item of fig. 8.

Apart from the foundation of existing building blocks – which already exceed the risk types described in this paper – the framework is generally open for extension by additional blocks, in order to implement further application-specific risk types.

One generic `GeneralizedDeviationReporter` class generates the output representation for all a generalized deviation objects. The accompanying charts (like those shown in fig. 2 – fig.7) are automatically generated along with the textual simulation report. They were implemented using the JFreeChart package.

As a result, a concrete, easily utilizable generic framework for risk assessment has been provided for modelers of discrete event systems.

VI. CONCLUSION AND OUTLOOK

In this paper, a generalized deviation concept inspired by quantitative finance has been introduced. Standard risk metrics like *Volatility* or *Downside Risk* have been deconstructed into five sub-function components for reference state generation, reference selection, penalization, normalization and re-dimensioning. The advantage of this approach is its flexibility, allowing modeling a wide range of

Deviations

Title	Recording	Risk Type	Reference	Selection	Penalty	Normalization	Re-Dimensioning	Value
Previous Minimum Deviation	Infection Rate	Runup	Previous Minimum	Nearest or Equal	Exponential Diverge	1/N	Logarithmical	0.51079
Deviation from Preferred Channel	Workload	Outside	Fixed Channel (80.0, 90.0)	Nearest or Equal	Linear Diverge	1/(|Valid Selections|)	Linear	4.90365
Transitions between Phases	Political Party A Support	Transition	Fixed Values (20.0, 40.0, 60.0, 80.0)	Nearest (Unequal)	Quadratic Diverge	1/(|Valid Selections|)	Square-Rooted	5.94093

top

Figure 8. Textual generalized *Deviations* section of DESMO-J's simulation report (cf. fig. 4, 5 and 7).

risk perceptions in numerous application fields of discrete event simulation. Risk conceptions like *Upside Risk*, *Outside Risk*, *Transition Risk, Critical State Risk*, *Drawdown Risk* or *Runup Risk* are easily expressible in the generalized deviation framework by adjusting its sub-functions.

It is proposed to complement descriptive standard statistics in discrete event simulation with generalized deviation metrics as introduced here, because GD offers inherent adjustable valuation standards for sophisticated risk estimation in various application fields, and thus facilitates the assessment of simulation dynamics for the experimenter.

The work described here is part of a more extensive approach to transfer quantitative finance concepts to discrete event simulation. Analog generalization of Value at Risk and Drawdown metrics, to analyze simulation dynamics extent resp. dynamics characteristics, will be described in future papers. This paper does not contribute to further development of risk metrics in their original finance context; but it serves as an example of the fruitfulness and potential of method transfer between different subject areas.

Giving an outlook, measuring risk by GD is canonically extendable by employing an analog *Generalized Potential* metric for assessing non-risky states, e.g. in safe state channels. In consequence, every simulation run could deliver one value pair $(risk, potential)$ per observed state variable, rating its dynamics (note that risk and potential are independent of cost and revenues, as explicated in sub-section III.A). By conducting a series of simulation experiments (for statistical confidence reasons), a series of $(risk, potential)$ pairs will be obtained, which may be visualized in two dimensional scatter plots per variable. Regarding these profiles, a quick overview of state variable characteristics may be gained, concerning welcome vs. undesired dynamics.

Another extension option is incorporating more than one state variable into generalized deviation. For example, higher level risk may consist of elementary risk factors, and overall risk may be better assessable by combining individual risk variables into a multi-dimensional state space, rated by a multi-dimensional penalty function.

Moreover, it seems desirable to also provide a time-weighted version of generalized deviation, as observations need not necessarily be equidistant in time. For this purpose, penalties have to be multiplied with the time span an observation is in force, compensated by the $normalize$ function, then dividing the weighted penalty sum by the total sum of penalized time spans.

Beyond the risk types delineated in this paper, further advanced application contexts are imaginable, e.g.

- *Path dependency*: As discussed in sub-section III.F, reference states need not be constant in time, but may depend on previously observed states. Beyond previous extremes, arbitrary reference functions are conceivable, generating e.g. moving averages of past states before s_i, linear regression of past states, spectral analysis with a moving window, etc.
- *Bandwidth dependency*: The upper and lower boundaries of reference state channels may be adjusted dynamically according to previously observed states, e.g. as a multiple of moving standard deviation. When combining bandwidth dependency with the above-mentioned path dependency, several technical analysis indicators from quantitative finance can be described as special cases of generalized deviation, e.g. *Bollinger Bands* [11], *Keltner Channels* [12] or *Envelopes* [13].
- *Time dependency*: Deviation may be computed with additional consideration of simulation time, e.g. to model windows of periodic system vulnerability. In this application case, simulation time of an observation may influence the reference, selection and penalty functions: reference states could follow a time-dependent function, or identical observed states may be penalized differently, depending on the time they were observed at.
- *Degree of Unpredictability*: Reference states may be determined by a prediction on basis of past observed states. In this scenario, the penalty function will measure accurateness of future state predictions.

Many more applications are conceivable. The flexibility of freely definable GD component functions opens up noteworthy potential for generic risk assessment, benefitting and possibly inspiring the range of discrete event simulation application fields.

References

[1] B. Page and W. Kreutzer, The Java simulation handbook: Simulating discrete event systems with UML and Java. Aachen: Shaker, 2005.

[2] J. Göbel, P. Joschko, A. Koors, and B. Page, “The Discrete Event Simulation Framework DESMO-J: Review, Comparison to other Frameworks and Latest Development,” in *Proceedings of 27th European Council for Modelling and Simulation*, 2013.

[3] B. Page, *DESMO-J (http://www.desmo-j.de)*. Discrete Event Simulation Framework: University of Hamburg, 2013.

[4] A. Koors and B. Page, “Transfer and Generalisation of Financial Risk Metrics to Discrete Event Simulation,” in *Proceedings of The International Workshop on Applied Modeling and Simulation 2012*, Genoa, Italy, 2012, pp. 100–107.

[5] A. Koors and B. Page, “Application and Visualisation of Financial Risk Metrics in Discrete Event Simulation: Concepts and Implementation,” unpublished.

[6] H. Markowitz, “Portfolio selection,” *The journal of finance*, vol. 7, no. 1, pp. 77–91, 1952.

[7] F. A. Sortino and R. van der Meer, “Downside risk,” *Portfolio Management*, vol. 17, no. 4, pp. 27–31, 1991.

[8] W. Harlow, “Asset Allocation in a Downside-Risk Framework,” *Financial Analysts Journal*, vol. 47, no. 5, pp. 28–40, 1991.

[9] B. M. Rom and K. W. Ferguson, “Post-Modern Portfolio Theory Comes of Age,” *Investing*, vol. 2, no. 4, pp. 27–33, 1993.

[10] S. Rachev, S. Ortobelli, S. Stoyanov, F. J. Fabozzi, and A. Biglova, “Desirable properties of an ideal risk measure in portfolio theory,” *International Journal of Theoretical and Applied Finance*, vol. 11, no. 01, pp. 19–54, 2008.

[11] J. Bollinger, *Bollinger on Bollinger bands*. New York: McGraw-Hill, 2002.

[12] C. W. Keltner, *How to make money in commodities*. Kansas City, Mo: Keltner Statistical Service, 1960.

[13] C. D. Kirkpatrick and J. R. Dahlquist, *Technical analysis: The complete resource for financial market technicians,* 2nd ed. Upper Saddle River, N.J: FT Press, 2011.