RZ 3599 (# 99609) 04/04/2005 (revised January 2006)
Computer Science 27 pages

Research Report

Cause-to-Effect Operational Risk Quantification

Chonawee Supatgiat, Chris Kenyon, Lucas Heusler

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Revised version

Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

Cause-to-Effect Operational-Risk Quantification and Management

Chonawee Supatgiat, Chris Kenyon, Lucas Heusler
IBM Zurich Research Laboratory

Abstract

Operational-risk quantification has recently become important owing to the new Basel II regulations. Current methods based on observation of losses and their magnitudes to quantify operational risk do not preserve the cause-to-effect relationship that shows how operational risk can be reduced, managed, and controlled. We introduce a cause-to-effect operational-risk modeling methodology that enables operational risk to be reduced, managed, and controlled. As part of this methodology we develop a decomposition algorithm to address the complexity of large-scale models. We demonstrate the use of this methodology with an example inspired by the settlement process of an inter-bank financial clearinghouse.

Introduction

Operational-risk quantification has recently become important owing to the new Basel II regulations that come into effect in the 2006-7 timeframe. These regulations require capital allocation for operational risk, complementing the existing requirements on market and credit risk. Many researchers and practitioners have proposed methods based on observation of losses and their magnitudes to quantify operational risk. However none of these preserve the cause-to-effect relationship that reveals how operational risk can be reduced, managed, and controlled. From a practical point of view, these are crucial aspects of operational-risk management and can only be directly addressed through cause-to-effect modeling.

We use the term cause-to-effect operational-risk quantification to describe models that capture the causes of operational failures and their resulting effects in terms of losses. This quantification includes explicit modeling of the linkage between cause and effect.

Cause-to-effect operational-risk modeling is important firstly for managing the operational risk beyond simple quantification. That is, if a financial institution wants to change the capital allocation required under Basel II for operational risk it must understand the root causes of operational risk and how they lead to loss events. Beyond this overall management of operational risk, cause-to-effect modeling also enables the inclusion of operational risk in the business decision processes, such as business process re-engineering, infrastructure re-engineering, and infrastructure operation.

In this paper, we introduce a new dynamic cause-to-effect operational-risk quantification methodology that shows how to model failure dependencies, impact dependencies, and which information to collect and monitor. We also introduce a new decomposition algorithm to break the complex large-scale problem correctly into smaller submodels without losing the important cause-to-effect relationships.

Our methodology is dynamic in that it captures failure dependencies such as the fact that failure rates for many software components exhibit a transient increase following software upgrades. It also captures impact dependencies such as that failures during certain time periods, say weekends, incur no losses whereas the impact will be extreme if a failure of significant duration occurs during rush hour. The model also reveals a reduced operational risk on hardware replacement and when better maintenance policies are applied.

We demonstrate how cause-to-effect operational-risk quantification can be applied in practice using a simplified version of an inter-bank settlement process. This settlement process is based on a large financial clearinghouse whose operations are outsourced. Our goal is to show how our cause-to-effect methodology can be used to resolve risk-management questions. Specifically we examine one architectural question, i.e., the value of redundant systems, and one operational question, i.e. the value of different hardware replacement policies. The fact that the settlement operation we study is outsourced means that it is particularly easy to identify the value and the definition of business disruption. The definition of business disruption is specified in the service-level agreements (SLAs) and their value is defined, for the purposes of this investigation, by the penalty clauses.

Previous Work

Previous work ranges from high-level approaches, e.g. insurance, to the many recent loss distribution models.

High Level Approaches

Doerig (2003) provides comprehensive high-level suggestions for operational risk management best practices in the financial service industry. Operational risk can be mitigated by insurance. Operational risk insurance products are offered by, for example, Swiss Re's FIORI (Financial Institutions Operational Risk Insurance) product and AON's e-business risk insurance solutions (Doerig, 2003). In the area of business-disruption prevention, IBM Business Continuity and Recovery Services unit (IBM, 2004) provides solutions such as backup facilities or guaranteed emergency repair/replace services.

Methodology Classification

Doerig (2003) categorizes methodologies into three types: i.e. factor-derived or indicator-based; statistical / actuarial / simulation-based; and loss-scenario / qualitative assessment. Only the simulation-based models can capture the dynamics of the underlying system. With this type of model, users can see the impact or effects of

operational-risk management actions on the operational-risk numbers. In the financial industry, the loss-scenario methods are the most widespread.

Loss-Scenario Methods
Many researchers have worked on the estimation of loss distributions, which are key inputs to their risk assessment models. Medova (2000a, 2000b, 2001) discusses the use of extreme value theory in modeling the rare but catastrophic events of operational risk. Significant efforts in collecting loss data have been done by most banks (Ramadurai *et al*, 2004; Finlay *et al*, 2002).

Dependency Modeling
Wilson (1999) explains a causal approach in modeling the relationship between failure events. With this approach, the conditional dependencies among events are taken into account, making the input assumption realistic. Hageback et al. (2003) suggest a way to capture risk event dependencies by copulas. However, when being implemented in a real system, it requires numerous assumptions, e.g. the copula must be standard. Thus, the results obtained by this method run the danger of being less meaningful. Gewald et al. (2004) develop a framework to decompose operational risk into a matrix, and claim that it can be used as a Bayesian Belief Network in order to assess operational risk. The paper, however, does not provide supporting details of the claim. Leippold et al. (2003) provide a mathematical framework for modeling operational risk, but cause-to-effect relationships are not the focus of their paper.

Simulation Methods
Simulation is widely used in business modeling but has not yet been used to model operational risk explicitly. In Monte-Carlo simulations, the business is modeled and simulated using a discrete-event simulator. This enables the capture of the cause-to-effect relationships that are vital for understanding and managing operational risk – and not just quantifying the losses. Simulation modeling can be applied to most real systems. The complexity and size of real systems are one of the main practical challenges in operational-risk modeling. Some of the contributions of the methodology presented here relate to how to choose the appropriate level of aggregation at which to model and to a decomposition method to address the complexity issue. A standard simulation model will not be helpful if the modeler does not know how to decompose the big model correctly, such as we propose in this paper.

Cause-to-Effect Operational-Risk Quantification Methodology

In this section we explain the concept of model decomposition and the quantification methodology.

Model Decomposition Concept
The wide spectrum of operational-risk event types complicates quantification. Figure 1 shows our operational-risk failure-event taxonomy, which is based on the Basel II's classification of operational risk event types. There are more than 30 types of operational-risk loss events, and each type of event also has several subtypes.

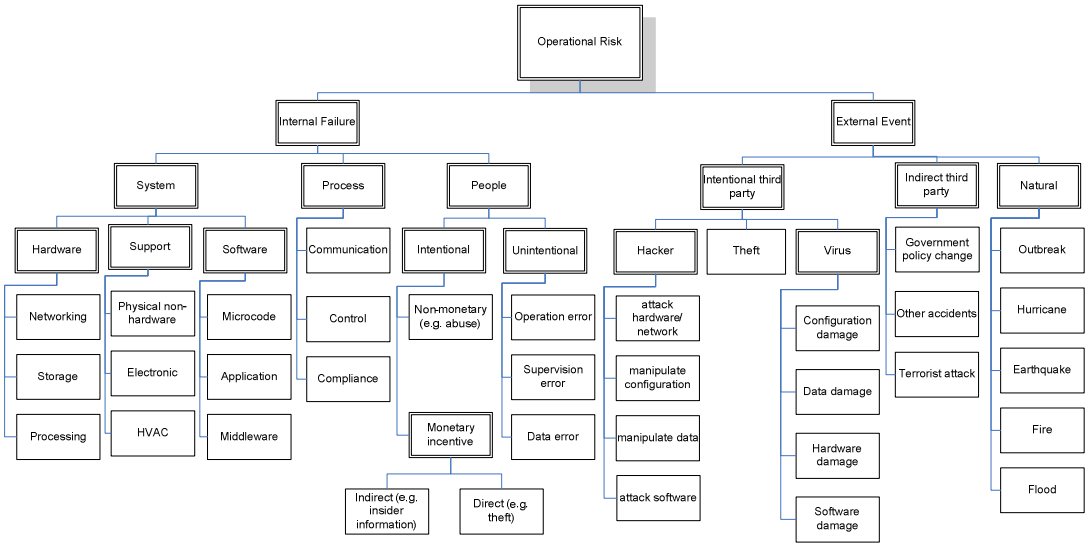

Figure 1: Operational risk taxonomy of failure-event types

When cause-effect relationships are modeled and all types of risk events are included, the model rapidly becomes complicated and intractable. Decomposing this large model into smaller submodels facilitates the modeling task, especially on highly complex systems typical of operational-risk quantification. We propose a decomposition approach to cope with this challenge. This approach maintains the failure and impact dependencies, thus facilitating the aggregation of the results in the final step.

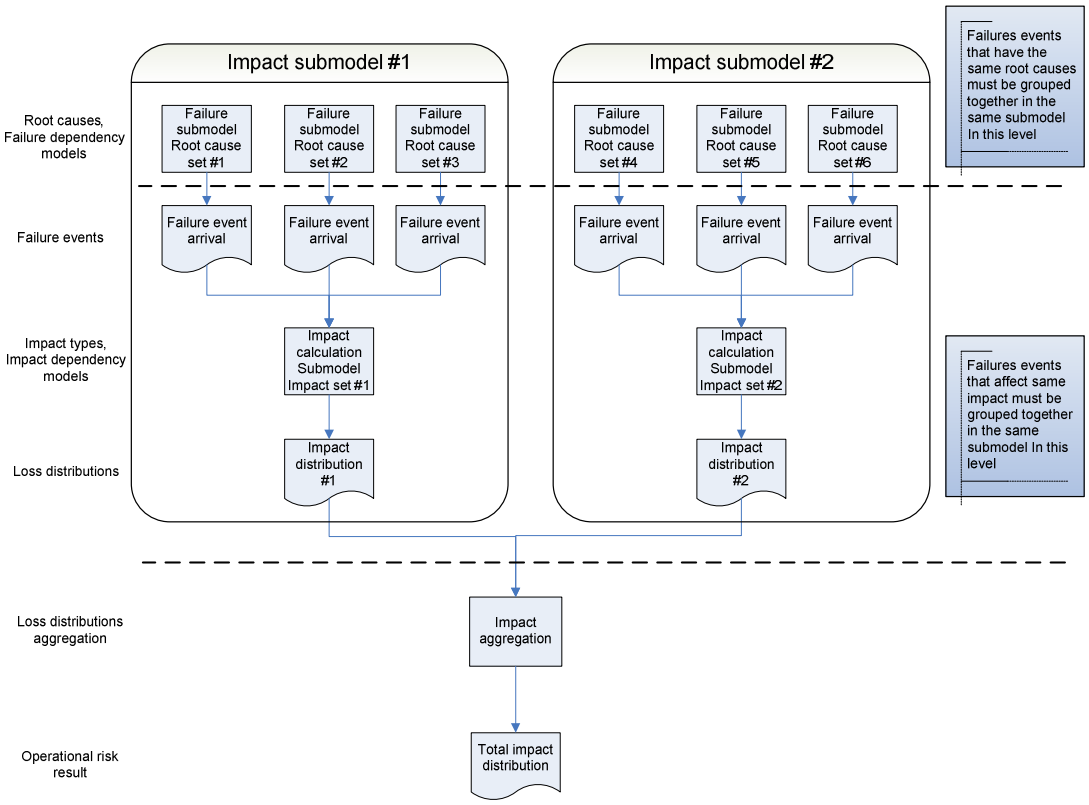

Figure 2: Concept of operational-risk model decomposition. Decomposition is first done by independent impact distributions. Impact distributions are built up from separate, unique root causes leading to failure events.

Figure 2 illustrates the concept of the *operational-risk model decomposition map* or *decomposition map*. The decomposition is done in two layers. In the first layer, the failure-event type occurrences from the Operational-Risk Failure-event Type Taxonomy are categorized by impact dependencies. The failure types that cause the same impact are grouped into the same submodel. For example, a contract violation penalty is calculated using the sum of the numbers of failure events of type A and B, thus both types must be in the same submodel in this layer. People stealing a company's secret would not cause a day-to-day business disruption impact; hence we can categorize such theft events into a different submodel than types of failures that entail a business disruption impact. We call the submodels in this layer the *Impact submodel*.

The submodels are decomposed further in the second layer. Within each impact submodel, failure-event types are categorized by their root causes of failures. The failure events that have the same root causes are grouped together in the same submodel. We call the submodels in this layer the *Failure submodel*.

The submodels in each layer can be dealt with separately. For each failure submodel, the modeler only needs to model the system in such a way that it generates correct

failure arrivals for each failure type. Because the failures from the same root cause can be correlated, having them in the same model allows us to correctly model the correlated failure arrivals.

For each impact submodel, the failures are translated into financial impacts. Since all failures having the same impact type are in the same submodel, the model can correctly calculate the resulting impact distribution. Moreover, the resulting impact distributions from different impact submodels are independent because they do not share root causes. Hence, these impact distributions can be easily and appropriately aggregated, i.e. by convolution, into the total impact distribution for the system. The convolution can be done numerically (or analytically if the outputs of the submodels permit it).

In general, the majority of the computation time is spent on solving the submodels rather than on the convolution. This is usually the case when submodels are simulation models. In this case, the benefit of the decomposition technique is to reduce the number of simulation replications. Suppose there are m submodels and each requires n replications, with this decomposition the complexity is $O(nm)$, whereas without this decomposition the complexity is $O(n^m)$.

In the following subsections, we describe our new cause-to-effect operational-risk quantification methodology based on this layering and decomposition concept. Figure 3 shows an overview of the workflow for the cause-to-effect operational-risk quantification methodology.

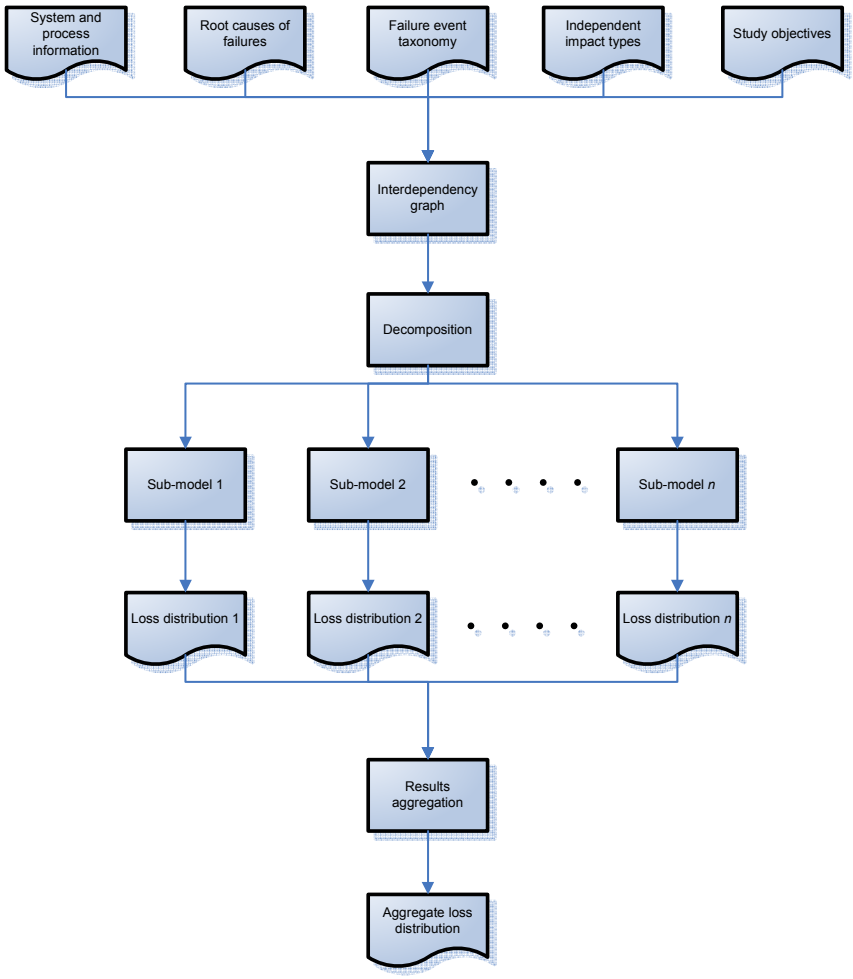

Figure 3: Cause-to-effect operational-risk quantification methodology.

Input information

The modeler needs to identify the study objectives of operational risk quantification and obtains the related system and business process information. The system and process information describes the business processes, people and IT systems of interest. This information is typically obtained from questionnaires and interviews as well as from process and IT architecture documents. The study objectives define the appropriate level of detail for the modeling.

Study objectives may be business process (BP) driven, information technology (IT) driven, or loss driven. Typical examples of objectives driven by these different interests are

- BP-driven objectives
  - What is the effect (in operational-risk terms) of adding a new insurance product to an existing people/systems infrastructure?
  - How can the operational risk of this BP be reduced by 50 percent?

- IT-driven objectives
  - What is the effect (in operational-risk terms) of consolidating these servers into one mainframe?
  - How can we reduce the operational risk of our database access by 50 percent?
- Loss-driven objectives
  - What are the three most important root causes of loss for this line of business and what would it cost to reduce them by 50 percent?
  - Should we have a mirror system for BP *X*?

The identified objectives drive the level of detail for model development, data collection, and monitoring.

Creation of the Interdependency Graph
To facilitate the decomposition, an *interdependency graph* is used. This graph, as shown in Figure 4, has three columns, namely, root causes of failures, failure-event types, and independent impact types. This links root causes of failure with failure-event types and their impact (loss distributions). An arrow from root cause R to failure event E denotes 'R can cause E', and an arrow from failure event E to impact type T signifies 'E can have impact type T'.

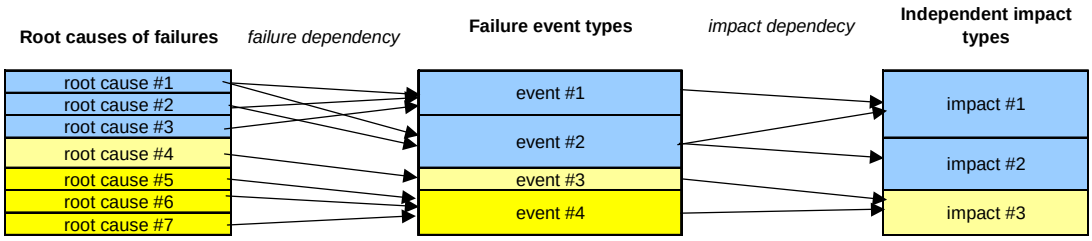

Figure 4: Interdependency graph linking root causes of failure with failure-event types and their impact (loss distributions)

The taxonomy in Figure 1 provides a general list of failure (or operational-risk) events to be used as a starting point for compiling the list of relevant risk events for the study. The impact of these risk events is also identified in this step, and failure and impact dependencies are determined. For example, in determining a failure dependency, the modeler needs to understand the relationship between the failure rate and the age of the failure component or the state of the system. In determining an impact dependency, he/she needs to understand the relationship between the impact and the system state or the failure duration. For example, it is very common that in a continuous impact, such as a business disruption, the impact increases — possibly exponentially — with the failure duration.

Base information to create the interdependency graph can be provided in many forms. Table 1, the *event-dependency chart*, is one such example that is used to assist modelers in identifying failure and impact dependencies. Typically, such an event-

dependency chart is derived from operations surveys, actual experience, and interviews

Most researchers and practitioners in operational-risk quantification currently do not have this step in their models. As a result, their models cannot capture the cause-to-effect relationships of the system.

Table 1: An example of an event-dependency chart

Sub-class	Events	failure arrival rate		Effect	Impact	
		f(age)	f(state)		f(duration)	Remark
Internal Hardware failure	hardware failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	backup system failover, replacement/repair of the failed hardware	yes	high if the backup also breakdown.Otherwise, low
	backup data storage failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	cannot retrieve history, recovering cost, replace/repair cost	yes	depend on the loss data and recovering time
	communication network failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	unable to communicate with customers, replace/repair cost	yes	depend on whether there is a critical information to convey

Decomposition using Interdependency Graph

The decomposition concept can be carried out appropriately by using an interdependency graph. In this step, we use standard graph-theory methods to identify disconnected (independent) subgraphs in the interdependency graph. For example, the graph in Figure 4 contains two subgraphs in the impact layer, i.e. the (1, 2, 3; 1, 2; 1, 2) subgraph and the (4, 5, 6, 7; 3, 4; 3) subgraph, where (x ; y ; z) denotes a subgraph containing root causes x, failure events y, and impact types z. These two disconnected subgraphs represent two separate impact submodels. As a result, the failure-event types that have the same impact are grouped together into one impact submodel. These impact submodels are shown in the first layer of the decomposition map in Figure 2

Within the second subgraph, there are two subgraphs in the failure layer, i.e. the (4; 3) subgraph and the (5, 6, 7; 4) subgraph, where (x ; y) denoted a subgraph containing root causes x and failure events y. These two disconnected subgraphs represent two separate failure submodels within their same impact submodel. As a result, the failure-event types that share the same root causes are grouped together into one failure submodel. These failure submodels are shown in the second layer of the decomposition map in Figure 2.

Solving Submodels

In this step, the modeler separately solves each submodel, i.e. each box in Figure 2, to obtain its output. For the failure submodels, the modeler needs to generate correlated failure-event arrivals or the failure probability distributions for each failure type. The common techniques to solve these submodels are statistical analysis and/or simulation. For the impact submodels, the modeler needs to translate the failure arrivals or distributions into impact distributions. The impact is represented in terms of monetary value, i.e. loss distributions. Depending on the complexity of the impact dependencies, the loss distribution might be deduced directly from failure distributions using an analytical approach; otherwise it can be done through simulation.

For each submodel, the system should be modeled at the *highest* detail level possible to avoid unnecessary work. For example, in a power-failure-event type, all components that share the same power line can be grouped into one single object in the model because they all will fail in a power outage. On the other hand, in a hardware-failure event type, each component should be treated as a separate object in the model because its failure pattern heavily depends on its age.

Results Aggregation
In this step, the impact distributions resulting from impact submodels are combined to obtain the net loss/impact distribution of the system. According to the decomposition, the resulting impact distributions from impact submodels are independent of each other because they do not share the same root cause of failure and their failure events do not affect the impact of the other submodels. Therefore, the impact distributions from the impact submodels can be correctly aggregated by convoluting them numerically. In a very special case, i.e. if all impact distributions are standard, it may possible to convolve them analytically.

Case Study Example

Overview and Objectives
To demonstrate the proposed methodology, we consider a simple version of a settlement process in an outsourced IT system for settlement processes in a financial clearinghouse.

The objectives of this example – beyond direct exposition and quantification – are to illustrate the use of this methodology for making business decisions impacting operational risk. Specifically, we examine a system-architectural question, namely, the value of having a redundant system, and an operational question, namely, the optimal frequency for server replacement.

All simulations were run for a five-year period using a discrete event simulation system (Arena™). Numerical results are given for the entire period.

The settlement unit consists of (potentially) two redundant systems. Each system comprises one server and one data storage unit. The business process and related IT map are shown in Figures 5 and 6, respectively.

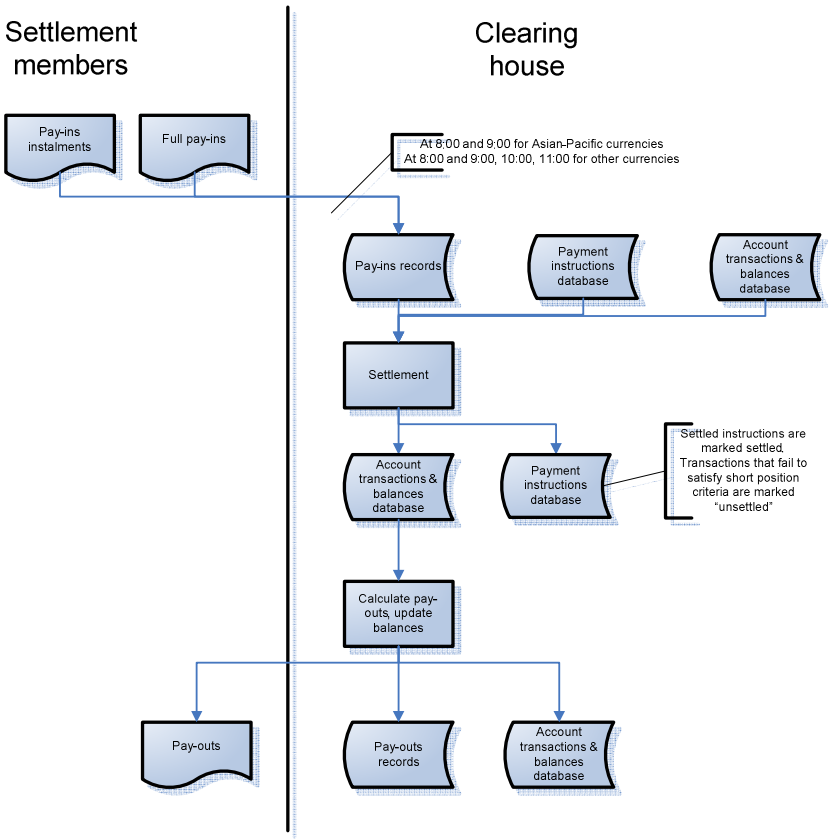

Figure 5: Settlement process

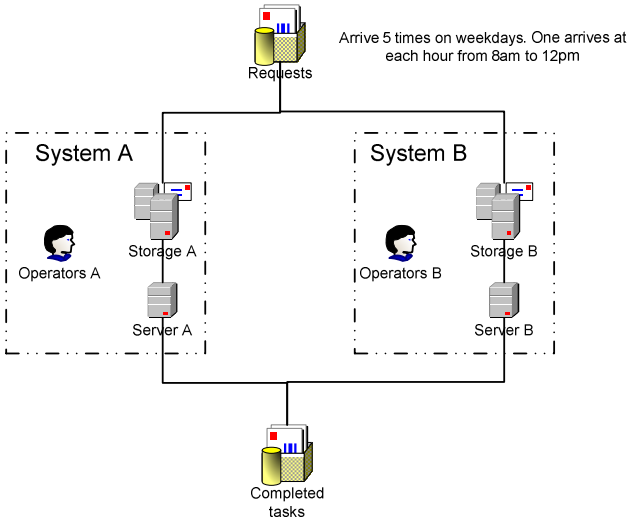

Figure 6: IT map of the settlement unit

For this exercise, let us assume that the practitioner wants to resolve two business problems. The first is an architectural problem, i.e. whether to have a redundant system, and the second is an operational problem, i.e. when to replace aging servers.

The level of detail needed in the model must be sufficient to capture the effects of these decisions.

Creation of Interdependency Graph
For this example, there are several types of impact. The most important one is penalty charge from SLA violation. The charge is calculated based on the performance and the breakdown time. The impact function is defined in a form of service credit (or service-level violation penalty).

Each month, the outsourcing service provider will be charged $500,000 if any one of the following events occurs:

- The aggregate breakdown time is more than two hours per calendar month.
- There is a single breakdown of more than one-hour duration.
- There is more than one breakdown of 30-minute duration or longer each month in the rolling three-month period.

Each month, the outsourcing service provider will also be charged $100,000 if any one of the following events occurs:

- The settlement completion is delayed by 30 minutes in any given day.
- It is delayed by ten minutes or more for two or more business days.
- The total delay time exceeds 90 minutes in that month.

Other impacts besides SLA violation penalties are: maintenance cost; disaster or other recovery cost; loss due to stealing of company assets or confidential information; and potential reputation loss, which includes the future sales loss.

The taxonomy in Figure 1 helps us in producing a list of potential operational-risk events related to this system. In the event-dependency chart in Table 2, this list is shown in the column labeled ‘Event’. The column *f(age)* explains the relationship between the failure rate and the age of the failing component (when all other state variables are fixed). The column *f(state)* of the failure arrival rate indicates the state of the world that influences the failure arrival rate. Similarly, the column *Remark* of the impact indicates the influence of the system state on the size of the impact. The column *f(duration)* indicates whether the impact size depends on the failure duration. Usually in a continuous impact, such as a business disruption, the impact increases — possibly exponentially — with the failure duration.

Table 2: Possible operational-risk events related to the settlement unit example shown in the *event-dependency chart*

Possible Operational Risk Events

Internal/ External	Main-class	Sub-class	Events	failure arrival rate		Effect	Impact	
				f(age)	f(state)		f(duration)	Remark
Internal	System	Internal Hardware failure	hardware failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	backup system failover, replacement/repair of the failed hardware	yes	high if the backup also breakdown.Otherwise, low
			backup data storage failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	cannot retrieve history, recovering cost, replace/repair cost	yes	depend on the loss data and recovering time
			communication network failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	unable to communicate with customers, replace/repair cost	yes	depend on whether there is a critical information to convey
		Supporting system failure	HVAC failure	high for new, low for non-new, high again for very old	high if bad maintenance	can lead to hardware failure, replace/repair cost	yes	depend on when it occurs, weekend or rush hours
			internal electricity system failure	constant over time	high if bad maintenance or a new change in electricity system	backup system failover	yes	high if the backup also breakdown.Otherwise, low
		Software failure	main settlement software failure	high for new version or new patch, low for old	high if there is a change in the system	backup system failover if detected	yes	high if not detect or the backup system is also breakdown.Otherwise, low
			non-core software failure	high for new version or new patch, low for old	high if there is a change in the system	non-core software mulfunction	yes	depending on how critical of the failed application
	People failure	Intentional	Stealing	higher for a new hire, decreasing over time	may depend on opportunities	monetary loss	no	mild (use company phone for private calls), high (steal company property)
			Sell customers' trade information to a spy	high for a new hire, decreasing over time	may depend on opportunities	reputational risk, leading to bankruptcy.	no	depend on the information
		Unintentional	operation error, e.g. acidentally switch off the server	high for a new hire, decreasing over time	same for every state	vary from no impact to business disruption	yes	depending on the error
			uninformed absent of an operator	independent of years of service	may be higher during a flu season!	no one operates the system	yes	high if the system require an attention
External	Third party	Intentional	Hacker/worm/virus attack	Increasing over time if no action is done to fix the volnerability	high when there is a new discovery of new volnerability	vary from no impact to business disruption	yes	high if not detect or the backup system is also infected. Otherwise, low
		Indirect	War	independent	higher if there is a tension with other countries.	vary from no impact to business disruption or total loss	yes	depending on the situation
			Terrorist attack	independent	higher if there is an evidence that it is a possible target for terrorists	vary from no impact to total loss	yes	depending on the attack
	Natural		Hurricane, Earthquake, Fire, Flood	independent	high if there is a weather/geology incident forecast	vary from no impact to total loss	yes	high if no warning or the backup system is also affected. Otherwise, low

All the events in the event-dependency chart (the ‘Event’ column in Table 2) are put into the middle section of the interdependency graph (the middle section of Chart 1). Based on the ‘failure arrival rate’ column in the event-dependency chart (Table 2), we can identify the root causes of the failures and drivers, and then list them in the interdependency graph (the left-hand section of Chart 1). Next we identify the impact types from the information in the ‘Effect’ and ‘Impact’ columns of the event-dependency chart and put them into the right-hand section of the interdependency graph.

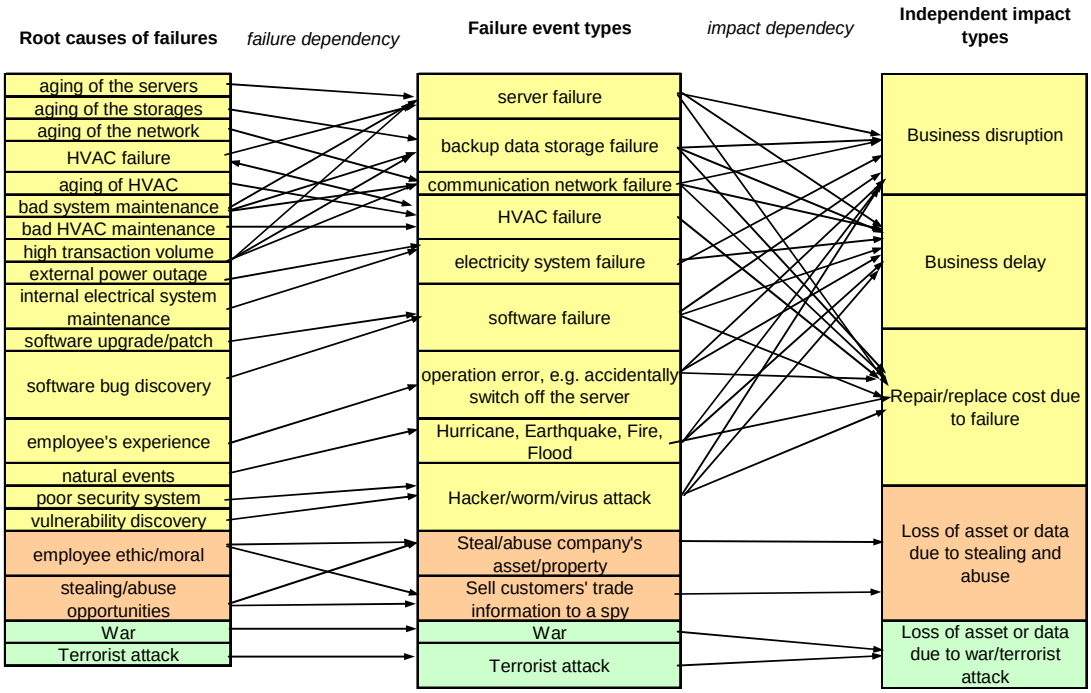

Chart 1: Interdependency graph

Finally, we link all three sections of the interdependency graph by arrows to indicate the dependencies. The information from the 'failure arrival rate' column of the event-dependency chart is used to create the failure dependency arrows, and the information from the 'impact' column is used to create the impact dependency arrows.

Decomposition using Interdependency Graph

Identifying the disconnected subgraphs, we break the graph in Chart 1 into three subgraphs, and mark them by three colors. Furthermore, the first subgraphs can be broken down into smaller disconnected subgraphs. The disconnected subgraphs in this layer are color-coded and shown in Chart 2.

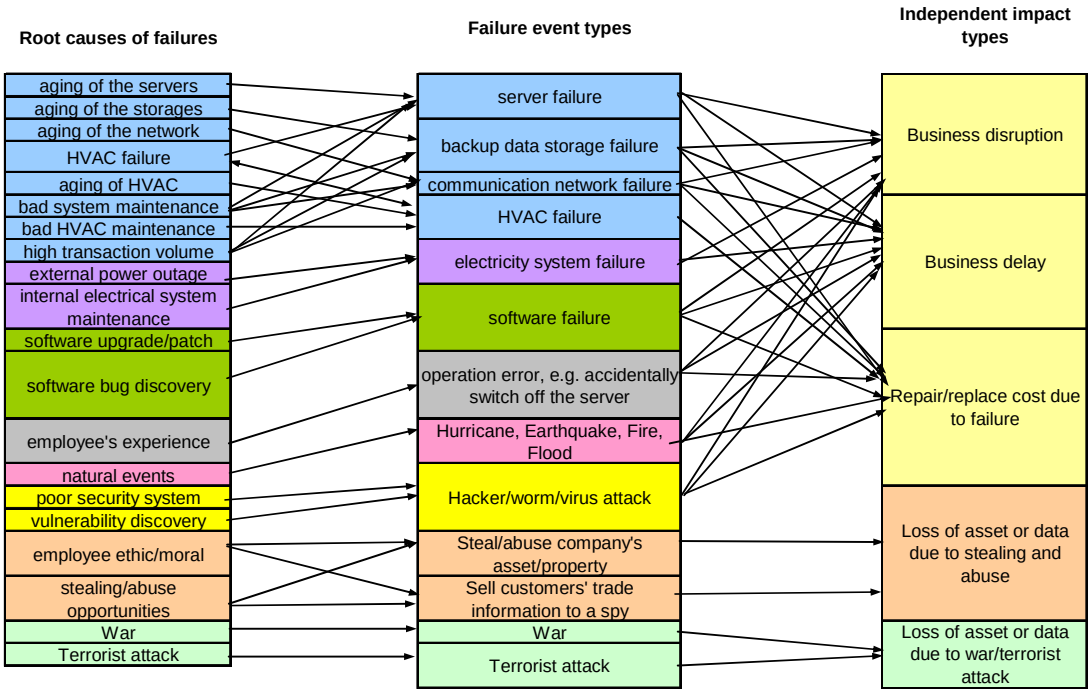

Chart 2: Interdependency graph decomposition

As a result, we have identified the decomposition structure for this problem. The decomposition map for this example is shown in Figure 7.

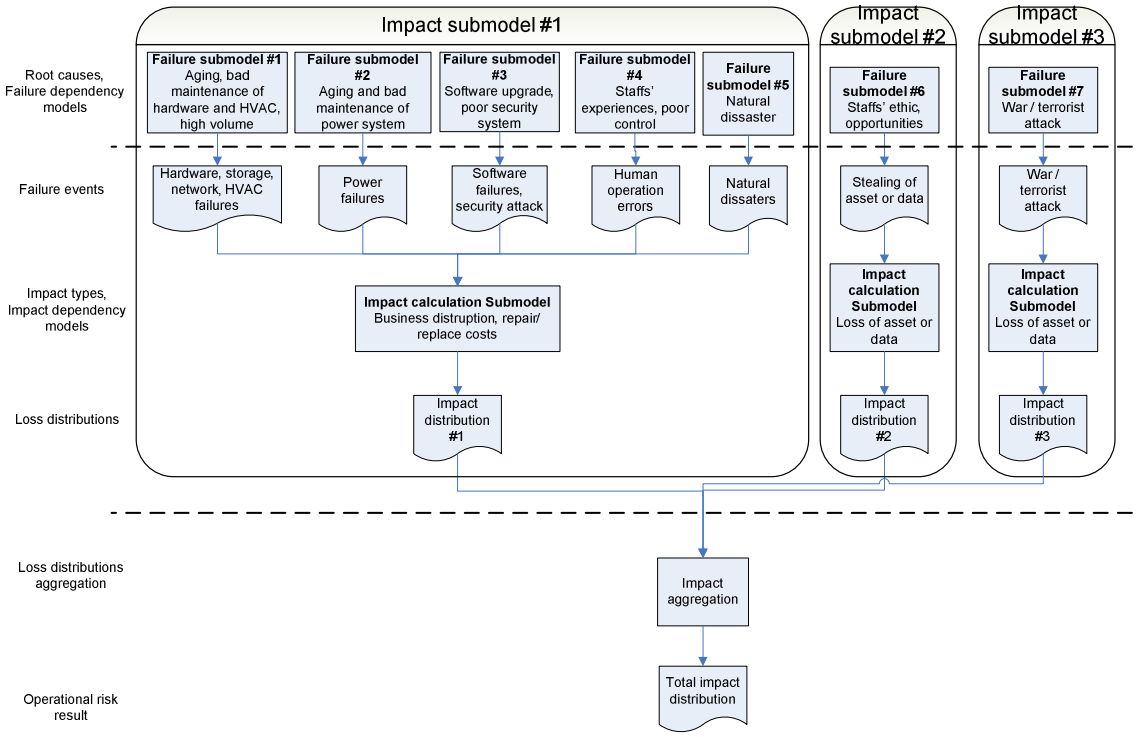

Figure 7: Decomposition map of this example

In this case, the business disruption impact is calculated by the SLA violation penalty described in the preceding subsection. The second layer in the decomposition map consists of three impact submodels. In this layer, all failure-event types that affect the same impact type are grouped together. In the impact submodel #1, the failures of hardware, power, software, human, and failures due to natural disasters can cause a business disruption or SLA violation. Therefore, they should be in the same impact submodel. The second impact type is the loss of assets or confidential data due to an insider, such as the legal cost and asset replacement cost incurred. We assume that operating assets cannot be stolen, whereas maintenance assets and spare parts can be. This type of event does not entail a business disruption, and hence can be assigned to another impact submodel. The same holds true in the case of war or terrorist attack, where the loss due to business disruption is protected by a force majeure clause in the SLA contract. The only impact of this event type is the costs of repairs and replacements.

Note that in this example we assume that a bad maintenance policy can affect the hardware, heating, ventilating and air conditioning (HVAC), and cause power failures. Human errors, such as accidentally switching off a server, have a direct effect in terms of business disruption, but no significant effects in terms of the hardware, HVAC and power failure rates. If we want to relax this assumption and allow human errors to affect hardware, HVAC, and power failure rates, then the failure submodels #1, #2, and #4 must be combined into a single failure submodel. In both cases, all the failure submodels can be practically implemented using a simulation approach.

System Modeling and Detail-Level Selection
For each submodel, the parameters to monitor are identified based on the root causes of failures and the failure and impact dependencies identified in Table 2. The submodel needs to contain detail levels such that those parameters can be monitored. The parameters and variables in the model are listed in the nonshaded columns of Table 3. The shaded columns are from the event-dependency chart constructed earlier.

Table 3: Failure and impact variables

Events	failure arrival rate		failure modeling	Effect	Impact		Impact modeling
	f(age)	f(state)	variables to monitor		f(duration)	Remark	variables to monitor
hardware failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	hardware age, transaction volume, current maintenance policy, HVAC state	backup system failover, replacement/repair of the failed hardware	yes	high if the backup also breakdown.Otherwise, low	backup system state, failure time, duration of breakdown, repair/replace costs
backup data storage failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	data storage age, transaction volume, current maintenance policy, HVAC state	cannot retrieve history, recovering cost, replace/repair cost	yes	depend on the loss data and recovering time	loss data, failure time, time to recover, repair/replace cost
communication network failure	high for new, low for non-new, high again for very old	high if high volume or bad maintenance	hardware age, transaction volume, current maintenance policy, HVAC state	unable to communicate with customers, replace/repair cost	yes	depend on whether there is a critical information to convey	information to communicate, failure time, time to recover, repair/replace cost
HVAC failure	high for new, low for non-new, high again for very old	high if bad maintenance	HVAC age, current maintenance policy	can lead to hardware failure, replace/repair cost	yes	depend on when it occurs, weekend or rush hours	failure time, time to fix, replace/repair cost
internal electricity system failure	constant over time	high if bad maintenance or a new change in electricity system	current maintenance policy and change in electrical system	backup system failover	yes	high if the backup also breakdown.Otherwise, low	backup system state, failure time, duration of breakdown
main settlement software failure	high for new version or new patch, low for old	high if there is a change in the system	age of the current version/service pack, change in the system	backup system failover if detected	yes	high if not detect or the backup system is also breakdown.Otherwise, low	fault activities performed, backup system state, failure time, duration of breakdown
non-core software failure	high for new version or new patch, low for old	high if there is a change in the system	age of the current version/service pack, change in the system	non-core software mulfunction	yes	depending on how critical of the failed application	fault activities performed, failure time, duration of breakdown

The failure and impact variables to monitor are the key to determine the level of details for each submodel. Each submodel should be at such a level of detail that these variables can be monitored.

In each impact-calculation submodel, the failures are transformed into impacts. The impact variables to monitor are necessary for impact calculation. For example, we can calculate the level of SLA violation due to a hardware failure if we know the state of the backup system, the failure time and duration, and the repair/replace cost. It is also possible that one type of failure can impact another type of failure. For example, a HVAC failure for an extended period of time can increase the failure arrival rate of some hardware components. Such correlated events can be captured by a simulation model, which is explained in the next subsection.

There are several different ways to estimate the failure or impact functions and their parameters. The most acceptable way is to perform statistical analysis of the historical data. If no historical data exists, the operational staff, who defines the dependencies listed in Table 1, should be able to at least 'guesstimate' or have a rough idea on the functions or their parameters. In the worst case, i.e. if we have no idea about a particular input assumption, a sensitivity analysis of that assumption must be performed.

Because in this paper, the input assumption modeling technique for each particular submodel is not our goal, we assume some dummy numbers, but that nevertheless make sense, for these functions and parameters in this example. For illustration purpose only, we now describe some of our input assumptions for this example.

Solving Submodels

Let us consider Impact submodel #1 in Figure 7. Here we see an example of cause-to-effect modeling in detail. The system failure submodel is complex, and many failures and impacts are interrelated, e.g. HVAC failure can affect the failure rates of other hardware components. Also, many impacts of hardware failures highly depend on whether the backup system is working.

The most important features included in Impact submodel #1 are listed below. We first describe the impact model, i.e. how failure events become losses, and then the failure submodels that generate the failure events.

Impact submodel #1

Impact (business disruption, business delay, repair/replacement costs)

- Business disruptions and delays incur penalties as defined by the SLA; these are important (costly) during business hours and not important (i.e. no penalty) outside business hours.
- Repair or replacement cost is stochastic and incurred depending on the hardware (including HVAC) failure severity.

Failure submodel #1

- Hardware aging-repair-replace: The hardware failure rate increases as the hardware gets older. For example, the mean number of days between server failures is equal to 1200 divided by the server age in months. The age of the hardware is reset to zero when it is replaced by new hardware.
- Facilities aging-repair-replace: The HVAC failure rate depends on its age.
- Utilization: Utilization of hardware can increase its failure rate; for example, the age of a storage disk increases by one month when it handles extremely high traffic or high volume. In each settlement round, the traffic can have an extremely high volume with a probability of 0.005.
- Knock-on effect of failures: If HVAC fails for an extended period, the hardware will be affected. For example, the hardware age increases by one month if the HVAC fails for more than one day.
- Queuing: The processing delay depends on the transaction volume as described by the queuing incurred. The delay increases when the volume increases (because of increased queuing).
- Redundancy: A failure in one system triggers a fail-over to another redundant system; if the redundant system is operational then there will be no business disruption impact.

Failure submodel #2

- Power outage: The arrival and duration of outages are random. The uninterruptible power supply (UPS) provides backup power for a maximum of one day.

Failure submodel #3

- Correlated upgrades: The model allows some of software upgrades to affect both systems at the same time. For example, 80 percent of software upgrades

will affect both servers, whereas 20 percent of the upgrades will affect only one server.

- Software upgrades: The failure rate of the software increases significantly when it is upgraded. Software failure includes all failure root causes, including security violations.
- Software maintenance: The software failure rate decreases when the software gets older. For example, the mean time between software failures is equal to $2*(\text{age in months})^2$.
- Software security: The effect of attacks on software is modeled by software failures and included in the software upgrades and maintenance items above.

Failure submodel #4

- Human error and experience: The operator error arrival rate is higher for a new-hire system operator. The possibility that a human error will affect both systems simultaneously can be different from the effects of software failure.

Failure submodel #5

- Natural disasters: Random arrival and severity.

Only simulation can handle this level of detail. We use Arena™ software to model and run the simulation.

Impact Submodel Results

In this subsection, we present example outputs of the impact submodels. In this particular exercise, we simulated 10,000 replications. The graph in Figure 8 is an output of Impact submodel #1, i.e. the model for business disruption, delay, and repair/replace cost due to failure. Table 4 shows the statistical results. The logarithmic graph (inset) makes it clear that there are no tail events of interest.

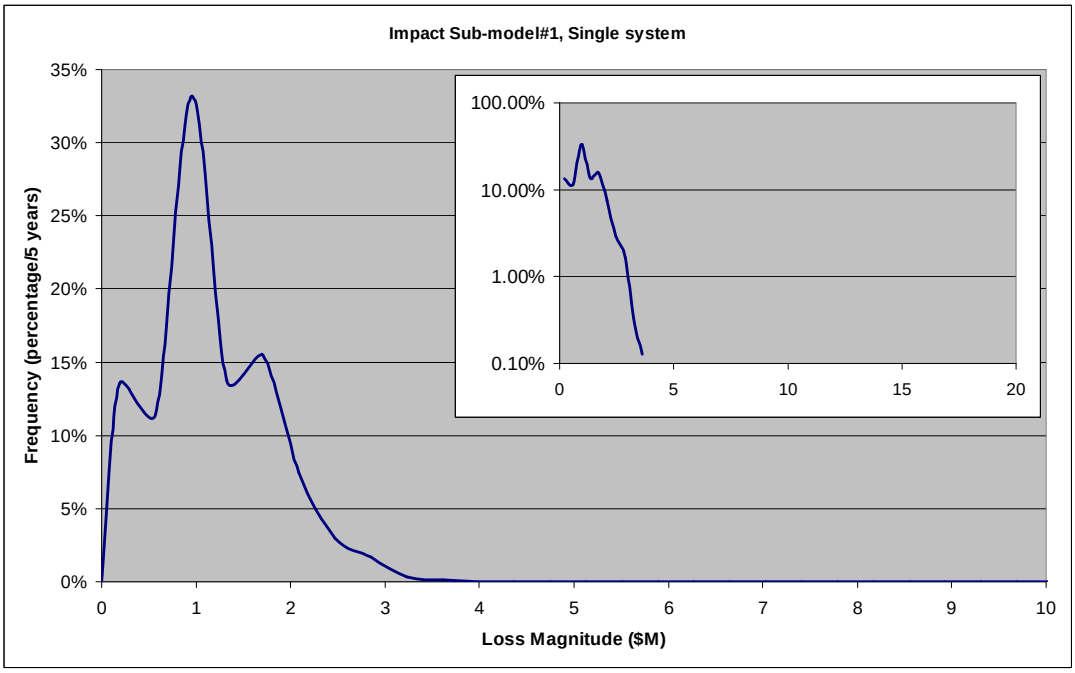

Figure 8: Loss distribution for business disruption, delay, and repair/replace cost due to failure. Linear and logarithmic (inset graph) plots shown with the same data.

Table 4: Statistical description of the loss distributions for business disruption with and without a redundant system. (For ease of comparison, the results for the redundant system are shown before the explicit consideration of redundancy in the text) Clearly the presence of a redundant system has a huge beneficial impact on business disruption/delay. We will see later that for the other impact submodels this is not nearly as dramatically the case.

	single	**double**
mean	1,144,811	397,922
s.d.	640,416	169,660
99% VaR	2,908,608	842,165
95% VaR	2,283,496	689,555
min	40	326
max	3,744,896	1,563,095

Figure 9 shows the loss distribution from Impact submodel #2, i.e. losses due to insider theft or abuse. Table 5 provides the statistical summary. Theft includes maintenance and spares as well as information. Again, no tail events are evident.

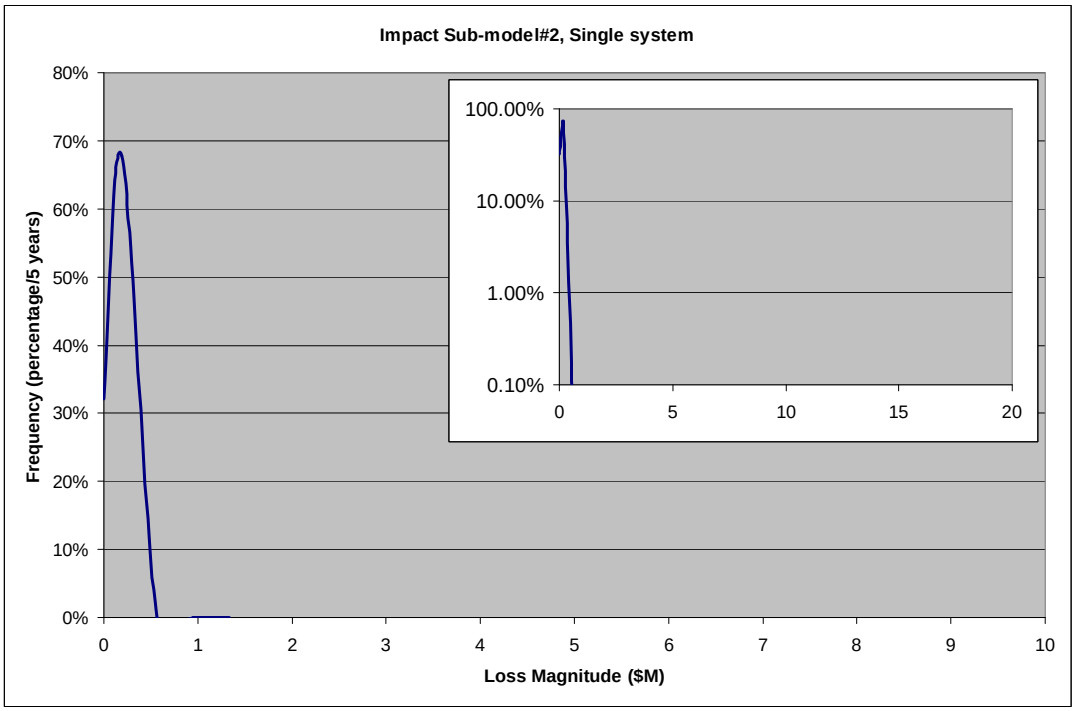

Figure 9: Loss distribution due to insider theft or abuse.

Table 5: Statistical description of the loss distributions for theft with and without a redundant system. With a redundant system, there is more to steal in terms of spare parts and maintenance supplies. However the overall differences are small.

	single	**double**
mean	15,759	23,709
s.d.	35,569	45,266
99% VaR	164,998	206,222
95% VaR	80,052	109,784
min	0	0
max	655,657	774,573

Figure 10 shows Impact submodel #3, i.e. losses due to war and terrorist attacks, and Table 6 gives the statistical summary. Here most of the events are in the tail and are due to both partial and total system losses.

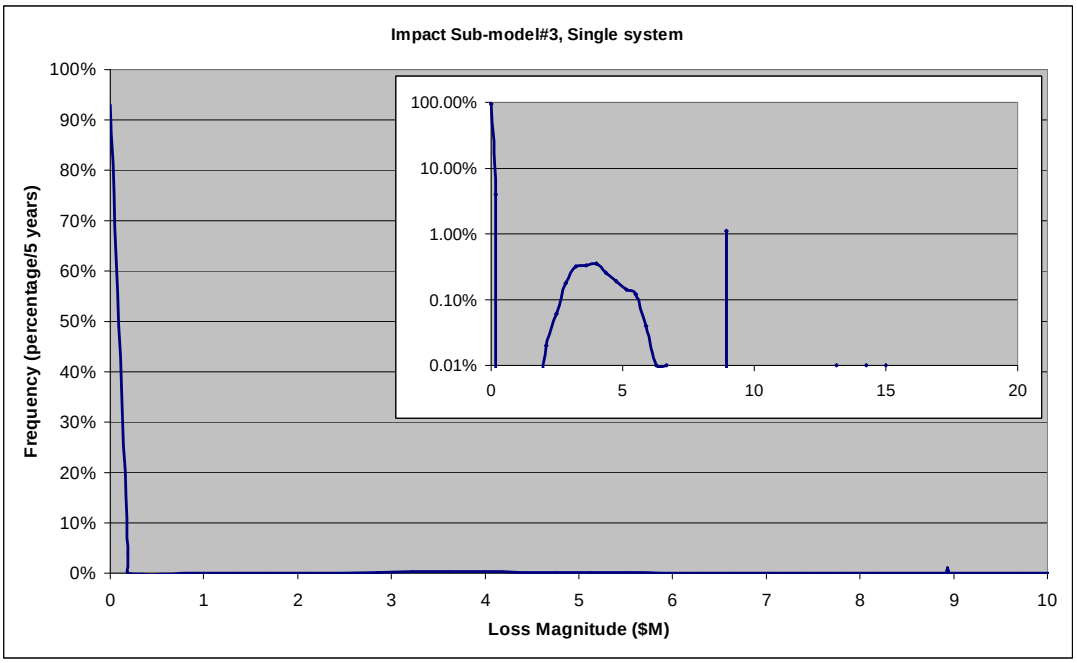

Figure 10: Loss distribution due to war and terrorist attacks.

Table 6: Statistical description of the loss distributions due to war and terrorist attacks with and without a redundant system. The redundant system incurs a slightly higher risk, which is due to the fact that the worst-case scenario, i.e. total loss, is actually worse in the redundant system than in the nonredundant system because there are more assets to lose.

	single	**double**
mean	182,959	204,331
s.d.	1,116,565	1,243,214
99% VaR	8,994,178	9,428,824
95% VaR	456	335
min	0	0
max	15,098,522	16,162,506

Result Aggregation

The independent loss distributions from the three impact submodels shown in the preceding subsection can easily be aggregated into the total loss distribution using a numerical convolution program. Figure 11 shows the total loss distribution resulting from aggregating the loss distributions of the three submodels for a nonredundant system.

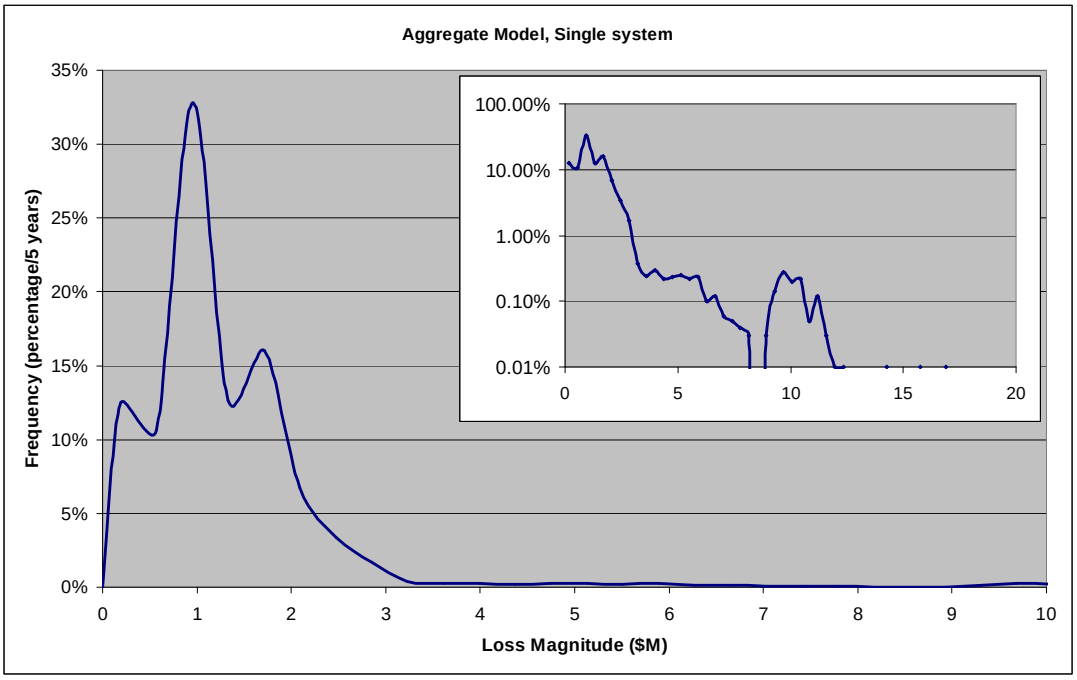

Figure 11: Aggregate (total) loss distribution obtained from the three impact submodels in the case of a nonredundant system. This is obtained by convolving the independent (by construction) impact distributions. See Table 7 for a statistical summary.

Some example results

Here we answer the questions we posed earlier, namely, what is the value (in operational risk terms) of having a redundant system; and what is the difference between different server-replacement policies. The former is a system-architectural question, and the latter is an operational question.

Redundancy system analysis

The total loss distributions in the two cases (with and without a redundant system) are compared in Figure 12 and Table 7. We see that although having a redundant system would reduce operational risk due to business disruption, it increases the operational risk due to insider stealing and war/terrorist attack, in addition to the extra cost of having a redundant system. For the decision on whether to have a redundant system, the decision maker should assess the risk preferences with respect to both mean and distributional aspects. In our example, although most indicators would clearly point to a preference for the redundant system, the single system might be preferable if the decision maker is only sensitive to the 99% VaR.

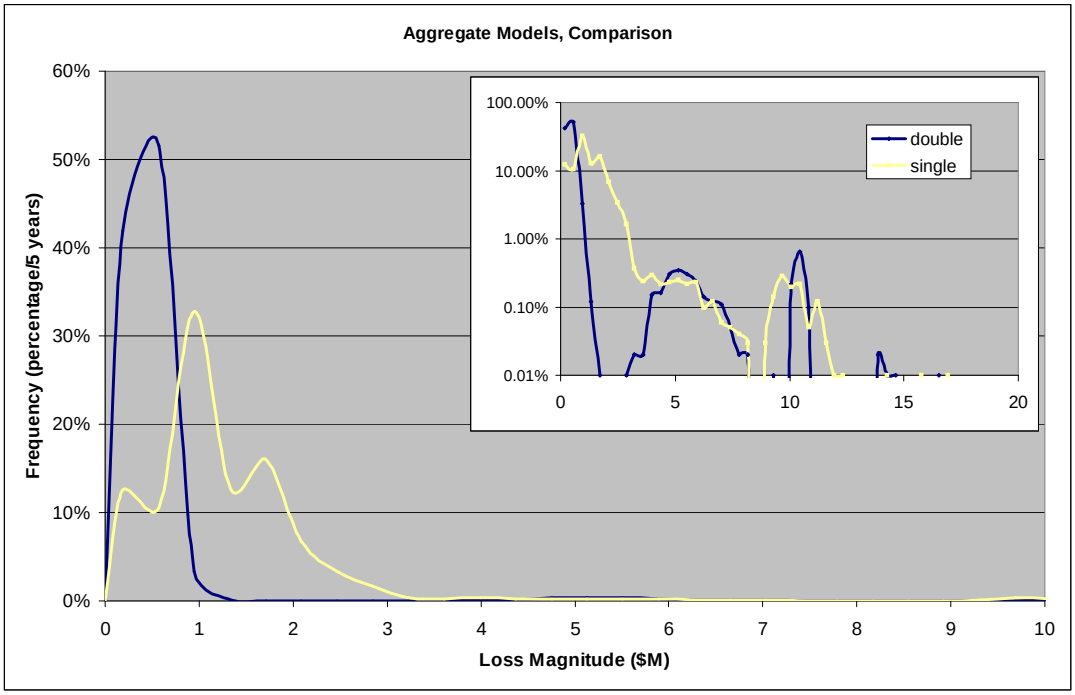

Figure 12: Total impact distribution with *(double)* and without *(single)* redundant system

Table 7: Statistical description of the total impact distribution with and without redundant system.

	single	**double**
mean	1,343,528	625,963
s.d.	1,292,178	1,256,480
99% VaR	9,247,169	9,906,728
95% VaR	2,740,314	819,350
min	40	326
max	19,499,075	18,500,173

The operational cost in the redundant system has a significantly lower mean than that in the single system. Regarding the business decision, especially for a large company, we found that if the initial investment cost for having the redundant system is lower than the difference between the mean of the two cases (roughly $700,000 over the five-year period used), the company should go for the redundant system.

Replacement policy analysis

We consider a range of replacement policies in a nonredundant system. We consider server replacement policies varying between 10 and 60 months. Figure 13 gives the resulting expected total losses when implementing these various replacement intervals. The policy with a 32-month replacement interval yields the lowest total expected loss.

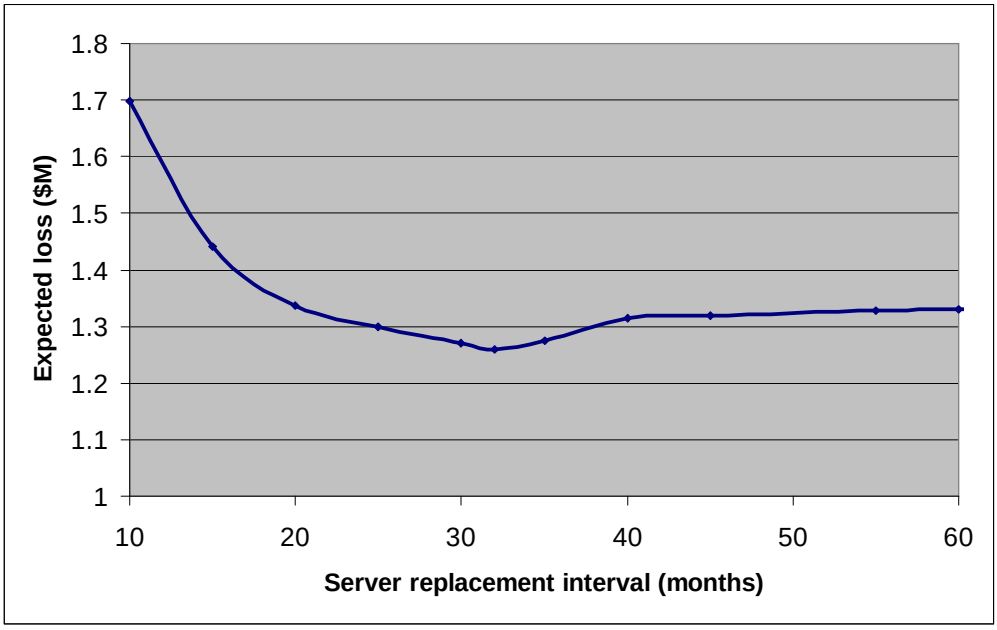

Figure 13: Comparison of different server-replacement policies in terms of expected total loss. Replacing the server when it reaches an age of 32 months will yield the lowest total expected loss.

Conclusion

Operational-risk quantification has recently become important because of the new Basel II regulations. Current methods based on the observation of losses and their magnitudes to quantify operational risk do not preserve the cause-to-effect relationship that shows how operational risk can be reduced, managed, and controlled. Cause-to-effect modeling is also necessary to capture the complex interdependencies of business processes and the systems supporting them. We have introduced a cause-to-effect operational-risk modeling methodology that enables operational risk to be reduced, managed, and controlled. To the best of our knowledge, we are the first to implement and use cause-to-effect modeling in operational-risk quantification.

We demonstrated the use of this new methodology with an example inspired by the inter-bank settlement process of a financial clearinghouse. We also introduced a new decomposition algorithm to address the complexity problem incurred when dealing with large-scale models. This allowed us to correctly separate large-scale models into smaller submodels without losing the important cause-to-effect relationships, and subsequently to aggregate the submodels.

In our example, we posed two questions regarding business decisions that potentially impact operational risk. One question was at the system-architectural level on the value of a redundant system and the other at the operational level on the effect of the server-replacement policy. We showed that, for our example, a redundant system was almost always preferable and that the policy of replacing the server when it reaches 32

months of age will yield the lowest total expected loss from all fixed-interval replacement policies.

The goal of this study was to demonstrate the methodology and how it can be used for answering operational-risk management questions. We showed examples in which operational risk quantification can be included in business-decision making both with respect to system architectural questions and operational policies while covering the entire range of people, process and system risks specified by Basel II.

Acknowledgement

The authors would like to thank Bala Ramachandran, David Gamarnik, Birgit Pfitzmann, Klaus Julisch, and Charlotte Bolliger from IBM Research for their helpful comments and suggestions.

References

Doerig, H.U. (2003) *Operational Risks in Financial Services: An Old Challenge in a New Environment.* Credit Suisse Group.

Finlay, M. and Kaye, J. (2002) *Emerging Trends in Operational Risk within the Financial Services Industry: Survey Report*. Raft International PLC.

Gewald, H. and Hinz, D. (2004) A Framework for Classifying the Operational Risks of Outsourcing: Integrating Risks from Systems, Processes, People, and External Events within the Banking Industry. *Working Paper*. E-Finance Lab. *http://www.efinancelab.de/*

Hageback, N. and Petz, L. (2003) Operational risk modeling: aggregating loss distributions using copulas. *Operational Risk*. Vol. 4, No. 7, pp 20-22.

IBM (2004) IBM Business Continuity and Recovery Services. *http://www.ibm.com/services/continuity*.

Leippold, M. and Vanini, P. (2003) The Quantification of Operational Risk. *Working Paper.* University of Southern Switzerland.

Medova, E. (2000a) Extreme Value Theory: Extreme Values and the Measurement of Operational Risk. *Operational Risk.* Vol. 1, No. 7, pp 11-15.

Medova, E. (2000b) Measuring Risk by Extreme Values. *Operational Risk Special Report, Risk Magazine.* pp 20-25.

Medova, E. and Kyriacou, M.N. (2001) Extremes in Operational Risk Management. *Working Paper.* Center for Financial Research, Judge Institute of Management, University of Cambridge.

Ramadurai, K., Beck, T., Scott, G., Olson, K. and Spring, D. (2004) Operational Risk Management & Basel II Implementation: Survey Results. *Special Report*. FitchRatings.

Wilson, D. (1999) Is Your Operational Risk Capital Adequate? Operational Risk supplement to *Risk Magazine*. Also at *http://www.financewise.com/public/edit/riskm/oprisk/opr-ibm.htm*