Risk-Aware Organizational Design

George Koliadis and Aditya K. Ghose

Decision Systems Laboratory
School of Computer Science and Software Engineering
University of Wollongong, NSW 2522 Australia
{gk56,aditya}@uow.edu.au

Abstract. Operational risk is an important, complex, and difficult, criterion to consider during any form of organizational decision making. In practice (1) complexity typically arises from: the use of a variety of risk related indicators; the use of multiple heterogeneous measurement scales; measurement uncertainty; varying levels of measurement precision; and, the widespread effect of each measurement; (2) difficulties arise due to the: time bound nature of the decision making process; and, the availability and interpretation of risk-related measurements. To help address these issues, we propose a conceptual framework to support and minimize the level of analyst involvement during the management of operational risk specified in organizational models. This is achieved by propagating/analyzing risk-related evaluations across descriptions of distributed, inter-dependant and mission critical work activities.

Key words: Operational Risk Management, Organizational Design

1 Introduction

Ad-hoc risk analyses are common within operational decision making [1]. The bounded nature of choice between alternatives requires time and information to be rationalized. As such, conceptual tools that help make use of, collect and disseminate available knowledge are critical. In many cases, precise risk assessment based on well defined problem specifications or statistical data are infeasible (e.g. in green-fields and time-bound projects). This may lead to increased levels of uncertainty with respect to the conditions surrounding some decision to be made. Efficient and effective techniques are required that allow for the combination of precise and imprecise assessments when evaluating risk.

In this paper, we outline a novel framework for propagating measurements (in a large class of measurement frameworks) across and between descriptions of inter-dependant work activities specified using a language influenced by the *i** [2] notation and the *c-semiring* [3] constraint modeling paradigm. This from of propagation allow us to: determine the effect of an evaluation across an organization; determine inferred bounds for improved analysis of organizational models; and determine when and where evaluations are inconsistent and adjust the evaluations or model to resolve an inconsistency. The examples we illustrate in this paper primarily refer to risk-related measurements, although the techniques we describe can also be deployed to model and analyze many other forms of measurements (e.g. cost, time, etc.). The theory outlined in this paper is implemented in the ISORROPIA Service Mapping Software Toolkit available for download at: *http://www.isorropia.org/*.

Section 2 provides a background and related work; Section 3 defines a risk modeling scheme; Section 4 describes risk propagation; we then describe how to deal with inconsistencies in Section 5; Section 6 describes some modes of analysis; and, we conclude in Section 7.

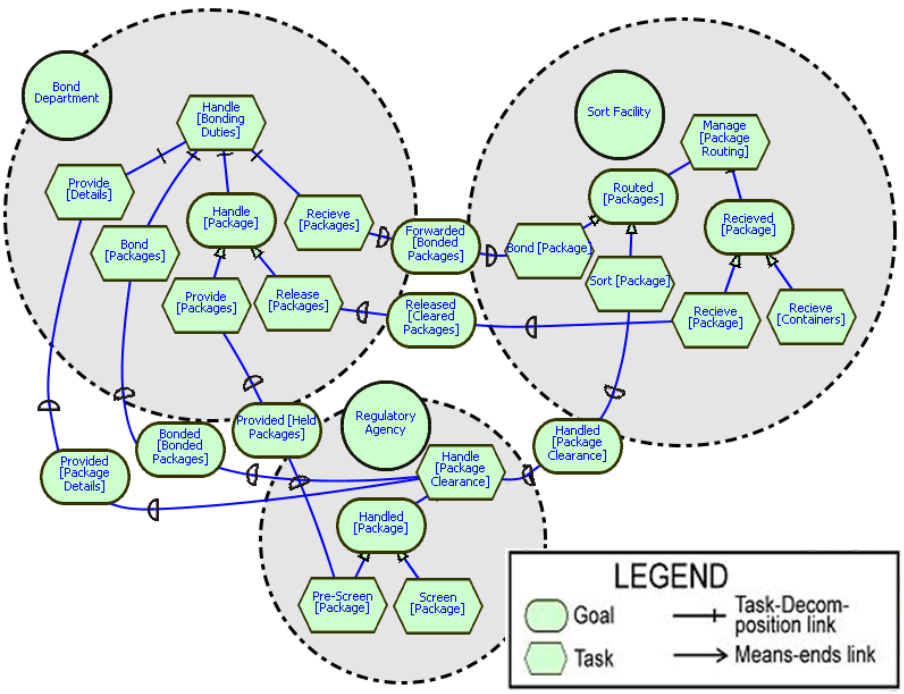

Fig. 1. An Organizational Model of a Transport Organization

2 Background and Related Work

Risk has been studied within many fields including economic theory [1], social science [4], project management [5], and software engineering [6]. In [7], risk is defined as "...exposure to a proposition of which one is uncertain" [7], whereby a self aware individual is *uncertain* of a proposition if they do not know it to be true or false, or it is unknown to them, and *exposure* is defined as the personal condition of that individual who *would* care (and has a preference toward) whether the proposition is either true or false [7].

2.1 Organizational Risk

In organizations, "...governance and decision making are the cause, risk and reward - the effect or outcome of those decisions" [8]; whereby, governance is defined as "...finding ways to ensure that decisions are made effectively" [9] by specifying "...the distribution of rights and responsibilities among different participants in the corporation, such as the board, managers, shareholders, and other stakeholders" [10]. Five core elements of risk include [11]: (1) Context, or the "...environment in which the risk is being viewed" [11]. The context describes the situation in which the conditions and consequences of a risk have some bearing, as well as identifying the scope of actions that trigger and can mitigate a risk. (2) Action, whose ramifications in certain conditions trigger a risk; (3) Conditions, when in combination with an action trigger a risk; (4) Consequences, or the potential effect of the action under certain conditions. These may lead to losses across many diverse attributes ranging from the stability of tasks through to the co-operation among work groups [12]; (5) Controls, to prevent, detect and correct the consequences of risk.

Operational risk is "...associated with the expected outcome of a process" [11], and is "enterprise-wide... endemic across the institution, affecting every business activity" [13]. [11] states that increased work distributions "...by their very nature increase risk", indicating a structural relationship. In organizational settings, risk becomes a key attribute for informing analysis and decision making that should be analyzed and controlled during organizational design.

2.2 Organizational Modeling

Organizational modeling in notations such as the *i** framework provide rich anthropomorphic abstractions for modeling the social, intentional, and strategic viewpoints of organizational actors. In *i**, dependencies such as *goals* to be satisfied, *soft-goals* to be satisficed, *task* to be performed, and *resources* to be furnished can be represented to help reason about the optimal delegation of responsibilities among actors in some organizational context. An actors' internal motivations and capabilities are represented as an AND/OR goal graph. In this mode, tasks may be *decomposed* and alternatives means for goals defined with *means-end* relationships. In addition, directed links between actors signify that a *depender* (i.e. source) actor depends on a *dependee* (i.e. target) actor for a *dependum* (i.e. the node between dependency links).

Take for example, Figure 1 - An Organizational Model of a Transport Organization. This model represents the interdependencies between three actors: a Bond Department; a Sort Facility; and a Regulatory Agency. Dependencies are represented as goals (ovals) connected to other types of goals (or tasks) via links labeled with a directional 'D'. For example, the Bond Department *depends* on a

Sort Facility *to achieve the goal of* “Forwarded[Bonded Packages]”. Within the scope of an actor, goals (or tasks) are decomposed into sub-tasks and goals or alternative tasks and goals (as represented in the legend in Figure 1).

In these types of models, risk and uncertainty are left implicit within the structure of an organization, and can sometimes be catered for within the underlying analysis framework - i.e. $i*$ incorporates a qualitative argumentation system for reasoning about levels of satisfaction towards soft-goals that has been extended [14] to cater for certain types of cancelation (relevant to risk mitigation).

2.3 Risk Modeling as Negative Preferences

Risk is measured by “...the probability of occurrence of loss/gain multiplied by its respective magnitude” [5]. Within the constraint satisfaction literature, similar issues relating to the uncertainty of a solution (or partial-solution) have required the use of soft-constraints, ranked according to probabilistic, fuzzy or weighted scales. In order to unify soft-constraint paradigms, [3] propose a abstract c-semiring framework that generalizes the classical (i.e. boolean), fuzzy, probabilistic and weighted approaches.

Definition 1 (Constraint-Semiring). A c-semiring (as in [3]) is defined as a 5-tuple $\langle A, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$ with the following properties: A is a set of abstract (numeric, boolean, or symbolic) preference values and $\mathbf{0}, \mathbf{1} \in A$; $\oplus$ is a binary operator used to compare preference values, which is closed (i.e. if $a, b \in A$, then $a \oplus b \in A$), commutative (i.e. $a \oplus b = b \oplus a$), associative (i.e. $a \oplus (b \oplus c) = (a \oplus b) \oplus c$), indempotent (i.e. if $a \in A$, then $a \oplus a = a$), has a unit element of $\mathbf{0}$ (i.e. $a \oplus \mathbf{0} = a = \mathbf{0} \oplus a$), and an absorbing element of $\mathbf{1}$ (i.e. $a \oplus \mathbf{1} = \mathbf{1} = \mathbf{1} \oplus a$); $\otimes$ is a binary operator used to combine preference values, which is closed (i.e. if $a, b \in A$, then $a \otimes b \in A$), commutative (i.e. $a \otimes b = b \otimes a$), associative (i.e. $a \otimes (b \otimes c) = (a \otimes b) \otimes c$), has a unit element of $\mathbf{1}$ (i.e. $a \otimes \mathbf{1} = a = \mathbf{1} \otimes a$), and an absorbing element of $\mathbf{0}$ (i.e. $a \otimes \mathbf{0} = \mathbf{0} = \mathbf{0} \otimes a$); $\otimes$ distributes over $\oplus$ (i.e. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$). In addition, the comparison operator induces a partial ordering $\leq_s$ such that forall $a, b \in A$ $a \leq_s b$ iff $a \oplus b = b$.

Risk measurement attempts to quantify the impact and uncertainty of some meaningful event. Traditionally, probabilistic and/or qualitative scales can be used. The *c-semiring* structure allows us to seamlessly describe and analyze precise and imprecise evaluations under the same unifying scheme. That is, qualitative evaluations do not need to be *massaged* into the same totally ordered numeric scheme.

Take for instance the case where two or more c-semirings have been used to model the problem at hand. In our instance, we would like to incorporate multiple risk measures (e.g. one risk measured using a numeric scale $\mathbb{R}^+$ and another measured using a symbolic scale $\{Low, Med, High\}$). This has been briefly discussed in [15] and more deeply discussed in [16]. The simple approach for combining two c-semiring instances produces another c-semiring instance (proved by [15]) that is the cartesian product of the abstract preference values, comparison operators, combination operators, and inclusion of least and most preferred values as respective tuples.

The *c-semiring* structure has been referred to as a *negative* preference structure in [17], used to compare and combine the values associated to tuples in the constraints of constraint satisfaction problems. This structure is a natural candidate for modeling risk measurements, as the combination operator monotonically decreases - i.e. the combination of two risks (occurring simultaneously) is worse than either of them occurring on their own. The structure also permits a *partial ordering* among values to allow for indecision among divergent evaluations.

Given the endemic nature of risk, the broad range of risk attributes (see [12]), and the application of risk assessments across many inter-related areas within an organization (e.g. financial, resource, project and software management), a combined, general and more holistic scheme would help to unify these otherwise independently structured and applied evaluations.

2.4 Related Work

It is surprising that little attention has been given to organizational risk analysis within the organizational modeling literature. For example, [2] discusses techniques for evaluating and analyzing actor *criticality* and *vulnerability* within organizations that may arguably attribute to both the *impact* and *likelihood* of certain classifications of failure. In other work, [18] discuss *obstacles* and [19] discuss *hazards* that may both obstruct the satisfaction of organizational goals, thus requiring analysis and mitigation. In addition, [20] discuss how quantitative risk evaluation and analysis can be incorporated into business process models to help evaluate taxonomic risk.

In [14] and [21], a goal-oriented and qualitative framework for modeling and reasoning about the consequences of risk (extending the formal framework of [22]) is presented. Speculative, hazardous and mitigating risk events and their contribution (positive and negative) towards organizational objectives can be clearly modeled. In [23], a goal-oriented, quantitative/probabilistic method for analyzing risks is presented. In their approach, risks are defined as fault-tree extensions to the existing NASA Defect Detection and Prevention (DDP) method.

The approach outlined in this paper aims to support mixed-mode and iterative evaluation of organizational structures with respect to a variety of heterogeneous endogenous (i.e. structural) and exogenous (i.e. environmental) risk measurements. As this task is knowledge intensive, we present a scheme to minimize analyst involvement by propagating evaluations across evolving organizational models. Our application is also primarily targeted toward risk analysis, although it could also be used in other settings. This permits improved visibility for risk related (and other) factors across an entire organizational model, even when the model is only partially evaluated.

3 Modeling Risk within Organizational Models

Risk evaluations are annotated to functional *goal* or *task* nodes on an organizational model. These evaluations choose values from a combined instance of a c-semiring risk scale. This evaluation provides a course grained measure of the risk of a deviation to the annotated node.

Definition 2 (Risk Evaluation). Let $R = \langle M, S, S_{low}, S_{upp} \rangle$ be a risk evaluation such that: $M \in \{Normative, Descriptive\}$ indicates the mode of an evaluation. $S = \langle A, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$ is a combined c-semiring scale we are using to measure the risk of a functional deviation, and this scale may be a composition of multiple scales/dimensions for each finer-grained deviation; S_{low} is a value from S (possibly an n-tuple) that indicates the lower bound on some evaluation, and states that the evaluation cannot get any *worse* than the designated value; S_{upp} is a value from S (also possibly an n-tuple) that indicates the upper bound on some evaluation, and states that the evaluation cannot get any *better* than the designated value; and, $S_{low} \leq_s S_{upp}$ (under the partial ordering $\leq_s$ induced by comparison operator $\oplus$).

3.1 Examples

The approach we have discussed for structuring risk evaluations can be used to capture and analyze *endogenous* (i.e. structural) as well as *exogenous* (i.e. environmental) risk under the same unifying scheme.

Exogenous risk exists outside the organizational model and is annotated (by an analyst) to describe the risk of a functional deviation. For example, a common scale attempts to measure risk as a real ($\mathbb{R}^+$) number. Let the risk indicator for some undesirable event be $\langle \mathbb{R}^+, min, max, \infty, 0 \rangle$, where:

- a value from $\mathbb{R}^+$ signifies the risk of a functional deviation measured as a positive real number;
- the better value when two risk values are compared ($a \oplus b$) is defined as the *min* of those values;
- the result of a combination between the two risk values ($a \otimes b$) is the *max* risk between them;
- the least preferred value is ∞; and,
- the most preferred value is 0.

Such a risk indicator could effectively be used where statistical information is available. For example, the risk of an untimely operation, where the timeliness of a result has been previously monitored. Another instantiation could be $\langle \{Low, Med, High\}, cp_I, cb_I, High, Low \rangle$, where:

- a value from {Low, Med, High} signifies the risk of a functional deviation;
- a comparison between two values will result in the best value given the ordering: High $<_s$ Med $<_s$ Low, where for any two $a, b \in A, a <_s b$ states that b is strictly 'better' than a, and $a <_s b$ iff $a \ cp_I \ b = b$;

- the result of a combination operation cb_I applied to two distinct values is the least preferred of the two;
- the least preferred value is High; and,
- the most preferred value is Low.

Endogenous risk manifests itself within an organizational model. For example, we could determine the vulnerability (or *dependence*) for a specific node as the set of agents ($a \in 2^A$) on whom the node depends. Let the vulnerability measure of a node be $\langle 2^A, \supseteq, \cup, A, \emptyset \rangle$, where:

- 2^A is the powerset of A;
- the comparison operator ($\supseteq$) is based on the superset relation, where for any two $a, b \in 2^A$, $a \leq_s b$ iff $a \supseteq b$ (the operator in this case would return the least upper bound in the lattice produced by this partial ordering);
- the combination operation ($\cup$) results in the union of the sets;
- the least preferred value is A; and,
- the most preferred value is $\emptyset$.

Another interesting endogenous indicator measures node *criticality* as the set of agents ($a \in 2^A$) that depend on a node. That is, if the node were to fail, the agents in the criticality set would be adversely affected. As such, criticality uses the same scheme as set forth for vulnerability. The difference lies in the local evaluations that provide a basis for propagation (i.e. we consider depender actors instead of dependees). The capability to capture and combine measurements from various risk indicators such as these is important.

4 Propagating Risk Evaluations Across Organizational Models

The evaluation of an element within an organizational model is situated within the context of all related nodes. For example, the evaluation of a task partaking within an AND decomposition (as a root or leaf node) also says something about the other related tasks. In order to gain a fuller evaluation of risk across an entire organizational model we provide some general techniques for propagating *bounded* evaluations bi-directionally across links within an organizational model.

The following constraints can provide a basis for propagation, defined with respect to alternative refinement patterns within organizational models:

1. $A_P \leq_s A_{C_1} \otimes \ldots \otimes A_{C_n}$ (parent A_P AND decomposed into $A_{C_1}, \ldots, A_{C_n}$);
2. $O_P \leq_s O_{C_1} \oplus \ldots \oplus O_{C_n}$ (parent O_P OR decomposed into $O_{C_1}, \ldots, O_{C_n}$);
3. $D \leq_s D_E$ (dependency D related to dependee node D_E).

Informally, the evaluation of: (1) the parent of an AND refinement can be no better than the combined value of its children; (2) the parent of an OR refinement can be no better than the best value among its children; and, (3) a dependency can be no better than the associated node realizing the dependency. These constraints, in combination with unary constraints for bounds, can provide a means to deploy constraint solvers in pre-processing or solving mode. Although, some types of operators and domains may not be completely catered to in traditional solvers (e.g. the vulnerability measures). The discussion in the following sections provide some lightweight heuristic guidance for adjusting bounds and resolving conflicts during change. Please note that organizational models do not typically include cycles.

4.1 Bottom-Up $AND \uparrow$ Propagation

Given the bounds S_{low}, S_{upp} of a risk evaluation for some parent node, and a set of bounds $\{S_{low_1}, S_{upp_1}, \ldots, S_{low_n}, S_{upp_n}\}$ for its children, we compute an $AND \uparrow$ propagated update of its upper (S'_{upp}) and lower (S'_{low}) bounds such that: $S'_{upp} = S_{upp_1} \otimes \ldots \otimes S_{upp_n}$; and, $S'_{low} = S_{low_1} \otimes \ldots \otimes S_{low_n}$; where $\otimes$ is the combination operator of the associated c-semiring scale. Note that evaluations of a dependency node from a dependee task propagate using this scheme.

4.2 Top-Down $AND \downarrow$ Propagation

Given the bounds S_{low}, S_{upp} of a risk evaluation for some child node, and the bounds of its parent S_{low_P}, S_{upp_P}, we compute an $AND \downarrow$ propagated update of its lower (S'_{low}) bound such that: $S'_{low} = S_{low_P}$ if $S_{low} \leq_s S_{low_P}$. Note that evaluations of a dependency node to a dependee task propagate using this scheme.

4.3 Bottom-Up $OR \uparrow$ Propagation

Given the bounds S_{low}, S_{upp} of a risk evaluation for some parent node, and a set of bounds $\{S_{low_1}, S_{upp_1}, \ldots, S_{low_n}, S_{upp_n}\}$ for its children, we compute an $OR \uparrow$ propagated update of its upper (S'_{upp}) and lower (S'_{low}) bounds such that: $S'_{upp} = S_{upp_1} \oplus \ldots \oplus S_{upp_n}$; and, $S'_{low} = S_{low_1} \otimes \ldots \otimes S_{low_n}$; where $\oplus$ is the comparison operator, and $\otimes$ is the combination operator of the associated c-semiring scale.

In this setting we treat the refinement as an inclusive evaluation of the parent node. We apply the comparison operation to the upper bound of all the children in order to determine the best possible value for the parent node. We then apply the combination operation to all the lower bounds of each child to determine the worst possible value for the parent node.

Alternatively, we could treat the refinement as exclusive and apply a pessimistic evaluation of each alternative. In this case, we apply a function to determine the worst case lower bound of each refinement. The parent node then inherits an evaluation of its upper and lower bounds from the child[ren] with this property. That is, $S'_{upp} = S_{upp_i}$ and $S'_{low} = S_{low_i}$ such that $i \in \{1, \ldots, n\}$ and $S_{low_i} = min(\{S_{low_1}, \ldots, S_{low_n}\})$.

Finally, we could have treated the refinement as exclusive and applied an optimistic evaluation of each alternative. In this strategy, we apply the comparison operator to the upper bounds of all child nodes and allow the parent to inherit the upper and lower bounds of the child[ren] with the best upper bound. Formally, $S'_{upp} = S_{upp_i}$ and $S'_{low} = S_{low_i}$ such that $i \in \{1, \ldots, n\}$ and $S_{upp_i} = S_{upp_1} \oplus \ldots \oplus S_{low_n}$.

4.4 Top-Down $OR \downarrow$ Propagation

Given the bounds S_{low}, S_{upp} of a risk evaluation for some child node, and the bounds of its parent S_{low_P}, S_{upp_P}, we compute an $OR \downarrow$ propagated update of its upper (S'_{upp}) and lower (S'_{low}) bounds such that: $S'_{upp} = S_{upp_P}$ if $S_{upp} \geq_s S_{upp_P}$; and, $S'_{low} = S_{low_P}$ if $S_{low} \leq_s S_{low_P}$.

4.5 Dependee and Dependency Propagation

Given the bounds S_{low_D}, S_{upp_D} of an evaluation for a dependency node, and the bounds S_{low}, S_{upp} of its dependee node, we compute a propagated update of its upper (S'_{upp_D}) and lower (S'_{low_D}) bounds such that: $S'_{upp_D} = S_{upp}$; and, $S'_{low_D} = S_{low}$. Propagation (or matching) of this kind may also occur in the opposite direction.

5 Dealing with Inconsistency During Propagation

Inconsistencies can be detected during value propagation. Given the current bounds of a risk evaluation S_{low}, S_{upp}, a propagation step (S'_{low}, S'_{upp}) can result in three types of outcomes:

1. **Bound Consistency.** The updated bounds are consistent with the current evaluation (i.e. $S'_{upp} \leq_s S_{upp}$ and $S'_{low} \geq_s S_{low}$). That is, a propagation should respect the current lower and upper bounds of a node, i.e. the result of a propagation should *tighten* an evaluation.
2. **Bound Inconsistency.** The updated bounds are inconsistent with the current evaluation (i.e. $S'_{upp} >_s S_{upp}$ or $S'_{low} <_s S_{low}$). A *relaxation* is one type of update that will inevitably be highlighted as an inconsistency. The problem here lies in accommodating the relaxation by resolving inconsistencies.
3. **Bound Incomparability.** One of the updated bounds is incomparable to a current bound (i.e. $S'_{upp} \oplus S_{upp} \notin \{S'_{upp}, S_{upp}\}$ or $S'_{low} \oplus S_{low} \notin \{S'_{low}, S_{low}\}$). The result of the $\oplus$ operator on incomparable values is their least upper bound (*lub*) in the c-semiring lattice.

Bound consistency does not necessarily mean that we can accept the update in a straightforward manner. This is due to situations where the updated bound is incomparable to the current bound. Dealing with these types of situations is discussed below.

5.1 Resolving Inconsistency

Inconsistency may be due to: mixed perspectives of multiple stakeholders / analysts; the timeliness of each evaluation where the newer evaluation should succeed an older evaluation; or even, identified when two incomparable values are provided. In any case, strategies are required to deal with such inconsistencies during propagation. Below we provide a short-list of possible strategies that may be applied, combined and/or integrated within the framework for risk analysis.

Cautious and Credulous. The *cautious* strategy selects the worst case for propagation among the set of alternatives. We can intuitively determine the worst case within our framework by applying the combination operator when evaluating the alternative to select.

Given two sets of bounds S_{low}, S_{upp} (current) and S'_{low}, S'_{upp} (updated) that are inconsistent, we define S''_{low}, S''_{upp} as the cautious revision of these inconsistent bounds such that: if $S'_{low} \leq_s S_{low}$, then $S''_{low} = S'_{low} \otimes S_{low}$, otherwise $S''_{low} = S'_{low}$; and if $S'_{upp} \geq_s S_{upp}$, then $S''_{upp} = S'_{upp} \otimes S_{upp}$, otherwise $S''_{upp} = S'_{upp}$; where $\otimes$ is the combination operator of the associated c-semiring.

The *credulous* approach is the inverse of the cautious approach. In this strategy, the analyst will choose to take the best case alternative with the hope that it will suffice. It is also just as simple to implement the credulous approach via the use of the comparison operator for each risk evaluation.

Given two sets of bounds S_{low}, S_{upp} (current) and S'_{low}, S'_{upp} (updated) that are inconsistent, we define a S''_{low}, S''_{upp} as the credulous revision of these inconsistent bounds such that: if $S'_{low} \leq_s S_{low}$, then $S''_{low} = S'_{low} \oplus S_{low}$, otherwise $S''_{low} = S'_{low}$; and if $S'_{upp} \geq_s S_{upp}$, then $S''_{upp} = S'_{upp} \oplus S_{upp}$, otherwise $S''_{upp} = S'_{upp}$; where $\oplus$ is the comparison operator of the associated c-semiring.

Model Re-Evaluation. Inconsistencies may reveal a need to re-evaluate an organizational model. For example, a normative evaluation on a dependency (e.g. any dependency in Figure 1) may become inconsistent with the descriptive evaluation on that node. If the measurements provided are valid, either: an evaluation may need to be relaxed; a goal, or goals, may need to be re-implemented (to achieve an acceptable evaluation); or, the model may need to be re-configured.

6 Analysing Organizational Risk

Under this framework the organizational model becomes a persistent source of knowledge that is required to bring together and combine distributed risk analyses during organizational decision making.

6.1 Modes of Analysis

The scheme we have discussed allows us to conduct the two broad types of analysis outlined below.

Analyzing Modal Evaluations. A local risk evaluation for a node on an organizational model may be either: normative - indicating the acceptable threshold for an evaluation (that may be propagated from another node); or, descriptive - indicating a current or future risk state within the organization. Distinguishing between normative and descriptive evaluations provides us with three important levels of analysis: at the level of normative measurement expectations; between descriptive and normative evaluations; and, between perceived descriptive evaluations. This is especially desirable when evaluating organizational models with multi-modal elements (e.g. the normative nature of dependencies).

Ordering Model Elements. Propagation also provides us with a basis for ordering elements on an organizational model, based on their bounded evaluations. Given two sets of bounds S_{low_i}, S_{upp_i} and S_{low_j}, S_{upp_j} for a specific risk, the status of their ordering may be either: strictly better ($S_{low_i} >_s S_{upp_j}$); conceivably better ($S_{upp_i} >_s S_{upp_j}$, $S_{low_i} \geq_s S_{low_j}$); incomparable, ($S_{upp_i} \geq_s S_{upp_j}$ and $S_{low_i} \leq_s S_{low_j}$). Ordering model elements in this way can help to identify areas of the model that are more prone to risk (i.e. a risk portfolio), and therefore requiring specific attention.

6.2 Evaluation and Propagation

Below, we illustrate two examples of evaluation and propagation. In Tables 1 and 2, we will use "$u|l$" to signify the *upper* and *lower* bounds of an evaluation.

Propagating Precise Risk Measures in Figure 1. Table 2 summarizes the local (L) and contextual (C) evaluations for one normative (i.e. L(N) and C(N)) and one descriptive (i.e. L(D) and C(D)) evaluation. The local evaluations describe the immediate evaluation of a node, prior to propagating the evaluation in order to contextualize the evaluation of other nodes in a model. Contextualized evaluations are inferred from the local evaluations, structure of the model and propagation procedure used. In this example, an initial $Normative$ evaluation at the $Manage[PackageRouting]$ node was provided, indicating that the aforementioned risk should only ever have a risk value within the range $[0, 0.01]$. The prominence of the node resulted in the evaluation being propagated to every node in the model (i.e. for brevity we have only included interesting evaluations in Table 1). Next, the $Recieve[Package]$ node was evaluated with a descriptive evaluation of 0.05, resulting in propagation to the $Released[ClearedPackages]$ and $Release[Packages]$ nodes. As a result, the descriptive evaluations at these nodes have been determined to be inconsistent with the normative evaluation, requiring resolution.

Propagating Vulnerability Measures in Figure 1. The vulnerability of a node is defined as the set of actors it depends on, including the owner of the node. In the setting we have described, local evaluation, propagation, and inconsistency resolution can be completely automated. Although the contextual

Table 1. Failure Likelihood of Figure 1

Node	L (N)	C (N)	L (D)	C (D)
Release[Packages]	$0\|\infty$	$0\|.01$	$0\|\infty$	$0\|.05$
Manage[Package Routing]	$0\|.01$	$0\|.01$	$0\|\infty$	$0\|\infty$
Receive[Package]	$0\|\infty$	$0\|.01$	$.05\|.05$	$.05\|.05$
Released[Cleared Packages]	$0\|\infty$	$0\|.01$	$0\|\infty$	$0\|.05$
⋮	⋮	⋮	⋮	⋮

evaluation of a node (as in the previous example) can be determined using other mechanisms (e.g. simple graph traversal), we would like to illustrate how these evaluations can be determined using the simple and general scheme for propagating evaluations that we have described. We start by approximating bounds. Naturally, the upper bound of each dependency (and related depender node) receive a value consisting of the depender and dependee actors. This indicates that these nodes are vulnerable to at least this degree. All other nodes receive an upper bound value consisting of the actor who owns the node. The lower bound on the other hand, may either: receive the entire domain of actors in the model - limiting the possibility for further propagation; or, receive the value of the upper bound as an strong approximation that is resolved using an inconsistency resolution strategy (e.g. the cautious strategy). Table 2 summarizes the results of the local evaluation and propagation of vulnerability across Figure 1 in the case of a strong approximation. For brevity we have only included a subset of the nodes and their evaluations.

Table 2. Vulnerability Analysis of Figure 1

Node	Local	Contextual
Handle[Bonding Duties]	BD|BD	BD,SF|BD,SF
Handled[Package]	RA|RA	RA|RA,BD
Manage[Package Routing]	SF|SF	SF|SF,RA,BD
Routed[Packages]	SF|SF	SF|SF,RA,BD
Sort[Package]	SF,RA|SF,RA	SF,RA,BD|SF,RA,BD
Received[Package]	SF|SF	SF|SF,BD
Handled[Package Clearance]	SF,RA|SF,RA	SF,RA,BD|SF,RA,BD
⋮	⋮	⋮

In this example, each node was assigned a crisp local evaluation indicating the immediate vulnerability of that node with respect to outgoing dependencies. For example, the $Sort[Package]$ task was given an evaluation of "SF,RA|SF,RA" since it is owned by the Sort Facility (SF) and depends on the Regulatory Authority (RA). The result of the propagation finally indicated that the $Sort[Package]$

and $Handled[PackageClearance]$ nodes are the most vulnerable in our Transport Organization.

7 Conclusion

Risk is an important consideration during organizational decision making, however there is little discussion of techniques to further support operational risk analysis in this setting. We outline a general framework for supporting risk assessment using rich organizational models. We provide an extensible means to define and incorporate highly configurable risk metrics for evaluation. The propagation schemes we have proposed, reduce analyst involvement over previous approaches, and allow for iterative and distributed evaluation driven by the detection of inconsistency. Furthermore, model elements can be ordered across specific risk-related dimensions to help in focusing attention to specific problem prone areas.

References

1. Knight, F.H.: Risk, uncertainty and profit. Hart, Schaffner, and Marx Prize Essays, no. 31. Boston and New York: Houghton Mifflin, Boston and New York (1921)
2. Yu, E.: Modelling Strategic Relationships for Process Reengineering. PhD thesis, Graduate Department of Computer Science, University of Toronto, Toronto (1995)
3. Bistarelli, S., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G., Fargier, H.: Semiring-based csps and valued csps: Frameworks, properties, and comparison. Constraints **4**(3) (1999) 199–240
4. Anderson, E.L., ed.: Risk Analysis - An International Journal. Blackwell Publishing (1981-)
5. Jaafari, A.: Management of risks, uncertainties and opportunities on projects: time for a fundamental shift. International Journal of Project Management **19**(2) (2001) 89–101
6. Boehm, B.W. DeMarco, T.: Software risk management. IEEE Software **14**(3) (1997) 17–19
7. Holton, G.A.: Defining risk. Financial Analysts Journal **60**(6) (2004) 19–25
8. Shaw, J.C.: Corporate Governance and Risk - A Systems Approach. John Wiley and Sons Inc, Hoboken, New Jersey (2003)
9. Pound, J.: The promise of the governed corporation. Harvard Business Review **73**(2) (1995)
10. Mathiesen, H.: The encyclopedia about corporate governance. http://www.encycogov.com (2006)
11. Alberts, C.J.: Common elements of risk. Technical report, Carnegie Mellon University, Software Engineering Institute, TN-014 (2006)
12. Gallagher, B.P., Case, P.J., Creel, R.C., Kushner, S., Williams, R.C.: A taxonomy of operational risks. Technical report, Carnegie Mellon University, Software Engineering Institute, TR-036 (2005)
13. Alvarez, G.: Operational Risk. Risk Books (2005)
14. Asnar, Y., Giorgini, P.: Modelling and analysing risk at organizational level. Technical report, University of Trento, DIT-06-063 (2006)

15. Bistarelli, S.: Soft Constraint Solving and Programming: a General Framework. PhD thesis, Computer Science Department, University of Pisa (2001)
16. Harvey, P., Ghose, A.K.: Relaxation of soft constraints via a unified semiring. In: Proceedings of the 2006 Canadian National Conference on Artificial Intelligence. (2006)
17. Bistarelli, S., Pini, M.S., Rossi, F., Venable, B.: Positive and negative preferences. In: Proceedings of the 7th International Workshop on Preferences and Soft Constraints. (2005)
18. van Lamsweerde, A., Letier, E.: Handling obstacles in goal-oriented requirements engineering. IEEE Transactions on Software Engineering **26**(10) (2000) 978–1005
19. Supakkul, S., Chung, L.: Applying a goal-oriented method for hazard analysis: A case study. In: Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications, Los Alamitos, CA, USA, IEEE Computer Society (2006) 22–30
20. zur Muehlen, M., Rosemann, M.: Integrating risks in business process models. In: Proc.eedings of the 16th Australasian Conference on Information Systems, Sydney (2005)
21. Asnar, Y., Giorgini, P.: Risk analysis as part of the requirements engineering process. Technical report, University of Trento, DIT-07-014 (2007)
22. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, J.: Formal reasoning techniques for goal models. In: Journal on Data Semantics. Springer Berlin / Heidelberg (2003) 1–20
23. Feather, M.S.: Towards a unified approach to the representation of, and reasoning with, probabilistic risk information about software and its system interface. In: Proceedings of the 15th International Symposium on Software Reliability Engineering (ISSRE'04). (2004)