RiskScape: A flexible multi-hazard risk modelling engine

Ryan Paulik (ryan.paulik@niwa.co.nz)
National Institute of Water and Atmospheric Research Wellington https://orcid.org/0000-0003-1147-6816

Nick Horspool
GNS Science Ltd

Richard Woods
GNS Science Ltd

Nick Griffiths
Catalyst IT Ltd

Tim Beale
Catalyst IT Ltd

Christina Magill
GNS Science Ltd

Alec Wild
National Institute of Water and Atmospheric Research

Benjamin Popovich
Moffat & Nichol

Glenn Walbran
Catalyst IT Ltd

Russel Garlick
Catalyst IT Ltd

Research Article

Keywords: RiskScape, Risk, Multi-Hazard, Multi-Hazard Risk, Software, Open Source

Posted Date: February 16th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1123016/v1

RiskScape: A flexible multi-hazard risk modelling engine

Ryan Paulik[1], Nick Horspool[2], Richard Woods[2], Nick Griffiths[3], Tim Beale[3], Christina Magill[2], Alec Wild[1], Benjamin Popovich[4], Glenn Walbran[3], Russel Garlick[3]

1 National Institute of Water and Atmospheric Research (NIWA), 301 Evans Bay, Greta Point, Wellington 6021, New Zealand

2 GNS Science, 1 Fairway Drive, Avalon, Lower Hutt 5010, New Zealand

3 Catalyst IT, Level 6/150 Willis Street, Te Aro, Wellington 6011, New Zealand

4 Moffatt & Nichol, 1780 Hughes Landing Boulevard, Suite 575, The Woodlands, TX 77380, USA

Corresponding author: ryan.paulik@niwa.co.nz

Keywords

RiskScape; Risk; Multi-Hazard, Multi-Hazard Risk; Software; Open Source

Abstract

This paper presents the architecture and features of RiskScape software. RiskScape is an open-source software with a flexible modelling engine for multi-hazard risk analysis. The RiskScape engine implements modeller-defined risk quantification workflows as ‘model pipelines’. Model pipeline steps and functions analyse hazard, exposure, and vulnerability data across different spatio-temporal domains using geoprocessing and spatial sampling operations. The RiskScape engine supports deterministic and probabilistic risk quantification, with several probabilistic-based modes described in this paper. RiskScape advances modelling software for multi-hazard risk analysis through several implementation features. The RiskScape engine operates model pipelines independent of system prescribed model input data classifications or standards. Multiple hazard types, metric intensities, and temporal occurrence information is geometry processed and sampled to create coverage data of simultaneous or sequenced multi-hazard events at object-exposure locations. Escalating multi-hazard event impacts are then determined for object-exposures using scripted conditional or nested statements that apply vulnerability functions in a logical sequence of temporal hazard and impact occurrence. These model features, supported by open geospatial consortium standard geospatial data files and operations, expedite RiskScape for modelling multi-hazard risk at any geographical location or scale.

1. Introduction

The RiskScape project develops modelling software for multi-hazard risk analysis. In this context, ‘multi-hazard risk’ refers to a system that analyses the occurrence and frequency of impacts from multiple hazard types and their associated phenomena occurring independently, in sequence or simultaneously (Kappes et al. 2012). The project aim is to address the well-publicised challenges of implementing multi-hazard risk analysis methodologies within modelling software (Delmonaco et al. 2006; Schmidt et al. 2011; Kappes et al. 2012; Komendantova et al. 2014). Achieving the project aim expedites modelling software as a tool to inform social and economic hazard risk research and disaster risk management in any global location (United Nations International Strategy for Disaster Reduction, 2015). To this end, RiskScape has progressed a flexible modelling engine for multi-hazard risk analysis.

RiskScape released a prototype desktop software application in 2012. The software operated system prescribed model workflows for single-hazard risk quantification across multiple hazard types (Schmidt et al. 2011). In 2015, the World Bank Global Facility for Disaster Reduction and Recovery (GFDRR) reviewed open-source and open access natural hazard risk analysis software. The review concluded that RiskScape prototype’s graphical user interface (GUI) and builder tools were “very easy to understand and a pleasure to use” but, software features were ‘quite simplistic’ and methods for vulnerability functions and calculating risk metrics ‘lacked transparency’ (Global Facility for Disaster Reduction and Recovery, 2015). Standardised model workflows in the prototype software offered limited support for modeller customised workflows and functions for different multi-hazard risk analysis contexts. In particular, risk quantification from single-hazards (e.g. single or multiple hazard metric intensities and their variability at a location) and their interactions and interrelations with multiple other hazards (e.g. tsunami following earthquake damage, extreme rainfall following wind damage etc.) were not supported by standardised model workflows. These issues led to the RiskScape prototype desktop software application being discontinued, with a renewed design and development focus to progress a platform-independent, transparent and configurable modelling software for multi-hazard risk analysis.

RiskScape software implements a model design, centred on a well-established conceptual framework for risk quantification:

$$R = f_C(H_i, E, V_i) \tag{1}$$

where risk (R) is a function (f_C) of the consequences from a hazard event (H) impacting an exposure (E) (i.e. element-at-risk). Consequences are determined from the exposure vulnerability (V) to an impact type and magnitude in response to either single or multiple hazard events (i). This framework is implemented in software that support model workflows for hazard risk analysis (e.g. Delmonaco et al. 2006; Schneider and Schauer, 2006; Schmidt et al. 2011; Cardona et al. 2012; Silva et al. 2014; Rossi et al. 2016; Aznar-Siguan and Bresch, 2019). Risk quantification principles are often similar between modelling software however, the implementation of model workflows and functions may differ. This is particularly relevant for risk quantification of 1) multiple types of single hazard events as independent scenarios; or 2) spatio-temporal interactions between multiple hazard events occurring at a location. While acknowledging principles and implementation approaches of other modelling software, several functional and non-functional requirements focus RiskScape software development for multi-hazard risk analysis (Table 1).

Table 1. Summary of RiskScape software development requirements.

Functional Software Requirements	Non-functional Software Requirements
• Calculate expected impacts from hazard events and quantify their uncertainties.	• Compatible with open-source software and software packages
• Provide a modular model framework for multi-hazard risk analysis	• Works on common operating systems, e.g. Windows, Linux, Mac
• Deterministic and probabilistic-based risk calculations	• Operable on multiple interfaces (i.e. command line (CLI), graphical user interface (GUI))
• Operate models in any geographical location	• Offline and/or online operational mode
• Calculate risk for spatial and temporal phenomena	• Scalability for use on a basic laptop, multi-tenanted, cluster or cloud computing
• Support a range of mathematical functions or relations to represent exposure vulnerability to impact from hazards (e.g. fragility functions, damage functions, damage-to-loss functions etc.)	• GIS file formats and geoprocessing functions are Open Geospatial Consortium (OGC) compliant
• Operate models using input and producing output data that are independent of data classification or standards	
• Interoperable with interpreted programming, languages, Geographic Information Systems (GIS), spreadsheet software, database software and other risk modelling software applications	

Model workflows and functions for multi-hazard risk analysis are critical functional requirements for RiskScape software. Model workflows must be modular and adaptable to support modeller configuration of workflow components. Several modelling software well-practised for single or multi-hazard risk quantification (e.g. HAZUS (Schneider and Schauer, 2006); CAPRA (Cardona et al. 2012); OpenQuake (Silva et al. 2014)), operate using standard model workflows or ‘calculators’ based on prescribed data classifications or standards.

RiskScape should operate model input data as independent entities enabling risk quantification for any hazard and exposure type combination, including spatio-temporal interactions. The challenge of multi-hazard risk analysis methodology implementation in modelling software is significant for researchers and practitioners, tasked with quantifying risk from complex hazard and exposure relationships, evolving under social, economic, and environmental changes (Komendantova et al. 2014).

This paper presents the architecture and features of RiskScape software, hereafter referred to as 'RiskScape'. RiskScape is open-source software with a flexible modelling engine for multi-hazard risk analysis. RiskScape is used in several previous modelling studies (e.g. Paulik et al. 2021; Williams et al. 2021; Woods et al. 2021); however, this paper is its first formal description. Here, the RiskScape engine software, system architecture and features are presented, with a focus on 'model pipelines' risk quantification. Model pipeline components and functions are described in the context of deterministic and probabilistic-based risk quantification, with emphasis on applications for multi-hazard risk analysis.

2. RiskScape Software and Implementation

2.1 Software

RiskScape is implemented using open-source Java programming language. Java is a platform-independent object-oriented programming language operated by numerous operating system software. Java's parallel processing meets the high computational demands for risk quantification at different spatial scales. Parallel processing also enables the consistent application of geospatial and statistical data processing operations in risk model workflows. Java supports geospatial data processing operations from open-source GeoTools (LGPL) and JTS Java libraries with open geospatial consortium (OGC) standard compliant file and database formats, webservices, coordinate reference systems, geospatial predicates and transformations.

RiskScape architecture centres on a core data processing 'engine' (Fig. 1). The RiskScape engine supports plugin functions that enable customised risk quantification workflows as a 'model pipeline'. A scientific code library of default functions (i.e. 'riskscape-defaults') perform geospatial, statistical, or other data handling operations in a model pipeline. Default functions are implemented using Java methods from GeoTools, JTS, Java Maths (i.e. 'java.lang.Maths'), and Apache Common Maths code libraries. These libraries are customisable, enabling default functions to be extended and implemented as independent objects within a model pipeline.

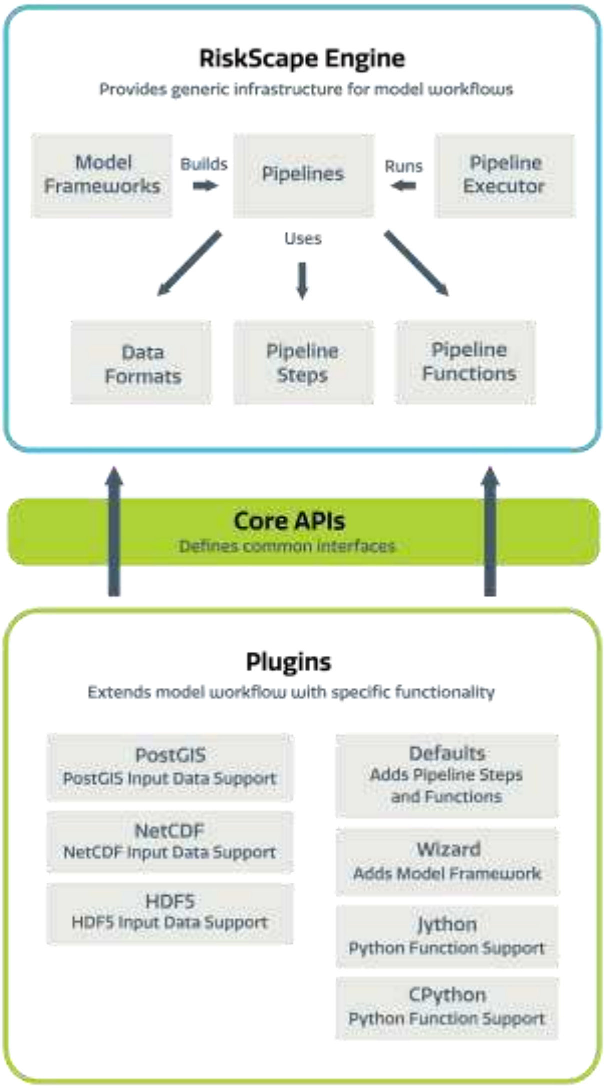


Fig. 1 A simplified conceptual diagram of the core RiskScape engine components and plugins.

The RiskScape engine supports additional programming languages as 'plugins'. The Python programming language supported by RiskScape engine default plugins. Python is an interpreted, high-level and general-purpose programming language supporting libraries (e.g. NumPy, SciPy) with extensive collections of mathematical functions. These features support mathematical functions or relations that represent vulnerability functions (see Section 2.2.5) in model pipelines. Here, vulnerability functions are engine independent and executed model pipelines using modeller-defined data classifications and logical expressions. This flexibility enables modeller-defined logic and conditions and relations for vulnerability function application to quantify exposure impacts from single-hazard or multi-hazard events. The RiskScape engine supports two default Python plugins for vulnerability function implementation. These include CPython, the C programming language standard Python implementation and Jython, a simple stand-alone Python version operating within the Java environment.

The RiskScape engine currently operates on a command line interface (CLI), implemented using the open-source Picocli Java library. The RiskScape engine CLI supports model pipeline configuration and execution. A RiskScape CLI Wizard plugin simplifies this process, while help documentation libraries (i.e. riskscape-i18n) support CLI operation. The plain-text help documentation can be modified to translate the RiskScape CLI for international languages or operate in different modelling domains. RiskScape use for research and development purposes is licensed with the Affero General Public License (AGPL) (GNU Operating System, 2007). The software is hosted on GitLab for use on Windows, Linux and Mac operating systems, track feature development and access scientific code documentation.

2.2 Implementation

The RiskScape engine implements the conceptual framework for risk quantification presented in Eq. 1. The engine implements the framework through model pipeline steps represented in Figure 2. Model pipelines, pipeline steps and step-functions supported by the engine for multi-hazard risk analysis are presented in Sections 2.2.1 to 2.2.7.

2.2.1 Model pipelines

The RiskScape engine executes model pipelines based on a domain-specific expression language that connects modeller-defined steps and step-functions. Data selection steps import (i.e. 'input data') and export (i.e. 'output data') to and from model pipelines. Steps where functions perform geometric transformations or numerical calculations based on input and output data selections include 'geoprocessing', 'spatial sampling', 'consequence analysis' and 'risk analysis'. The engine passes model input data between pipeline steps as 'tuples'. Each tuple represents a single data record or attribute list, such as an exposed object, e.g. building, road, farm etc. The tuple data structure is represented as a 'struct', a composite data type defining a grouped set of variables. Pipeline step-functions transform tuples based on math, geometry, logical and string operations that apply to the struct. Modified tuples are passed between each defined pipeline step.

Several configuration files support model pipeline step and data management. Text-based content files (.ini) provide a structure and syntax for the RiskScape engine to determine key-value pairs for properties and sections that serve multiple purposes organising and executing model pipelines. 'Model' files contain model pipeline data selection and step-functions while 'project' files define model input data files or directories, and data configuration files. 'Data bookmark' files store modeller-defined file paths and spatial data transformations that are set to locate and prepare input data for model ingestion. 'Data classification' files store attribute information representing model input and output data structs. Data classifications are modeller-defined and used in logical expressions or conditional statements to pair vulnerability functions with tuples. These associations can also be defined from other external files supported as engine plugins, e.g. the vulnerability function arguments and return type metadata defined in a Python file.

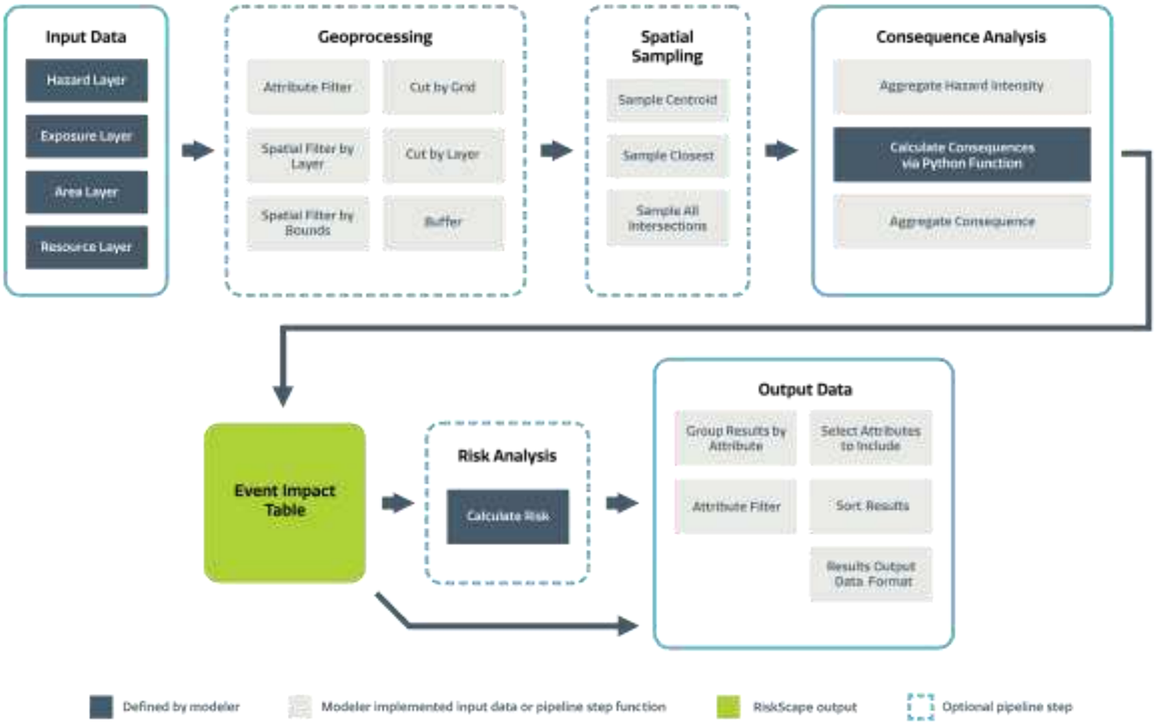


Fig. 2 A RiskScape model pipeline schematic representation of pipeline steps and functions.

2.2.2 Model input data

Model input data represents the core components for risk quantification presented in Equation 1. Required input data include 1) the 'hazard' event type, metric intensity, or temporal occurrence 2) the 'exposure' of objects (herein termed as 'object-exposures'), and 3) vulnerability functions (see Section 2.2.5) that determine object-exposure impacts to hazard type and metric intensities. Data 'layers' represent hazard and exposure geometry and attribute information in common GIS raster and vector data file formats (Table 2). Multi-dimensional data files including hierarchical data format (.hdf5) and network common data form (.nc), are formats supported as engine plugins for model pipeline ingestion of hazard occurrence (e.g. event probability, time-series) and metric intensity information. These file formats store many hundreds or thousands of hazard realisations, forming a stochastic hazard event set (SES) for probabilistic risk quantification (see Section 2.2.6). Optional input data include 1) ‘resource’ layers representing phenomena not classified in hazard or exposure layers but influences an exposures vulnerability to hazard impact (e.g. water table depth influencing liquefaction probability), and 2) ‘area’ layers that represent geographical areas for model output reporting of exposure impacts (see Section 2.2.7).

Table 2. A summary of model input data supported by RiskScape engine model pipelines.

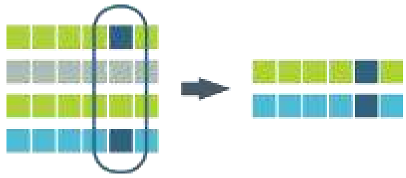
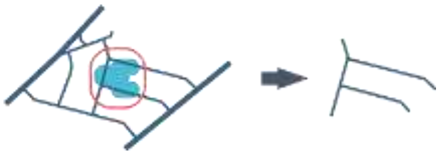
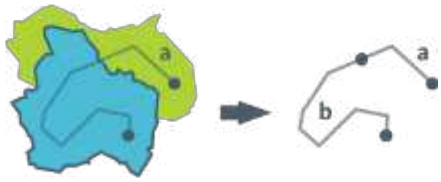
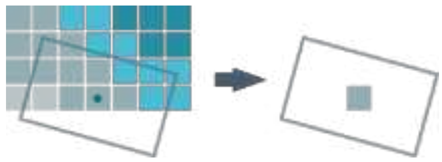
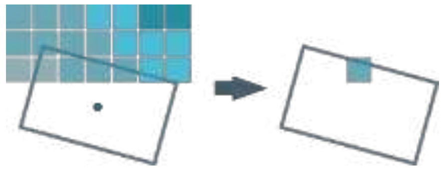
Input Data		File Formats	Geometry Types	Description
Layers	Hazard	.csv, .shp, .tif, .asc, .hdf5, .nc	Grid, Point, Line, Polygon	Required input data layers representing the geometries, spatial extent, metric intensity and/or frequency of occurrence for hazard events.
	Exposure	.csv, .shp, .tif, .asc	Grid, Point, Line, Polygon	Required input data layers representing the geometries and/or attributes of physical or non-physical exposures to hazard events.
	Area	.csv, .shp	Polygon	Optional input data layers to report object-exposure impacts from hazard events for geographical areas (e.g. suburbs, districts, regions).
	Resource	.csv, .shp, .tif, .asc	Grid, Point, Line, Polygon	Optional input data layers that represent phenomena not classified as a hazard or exposure but influence object-exposure vulnerability to impacts from hazard events, e.g. soil type.
Vulnerability functions		.py, .jar	NA	Vulnerability function is a general-purpose term for mathematical functions or relations that determine object-exposures impacts from hazard events e.g. fragility function, damage function, damage-to-loss function etc.

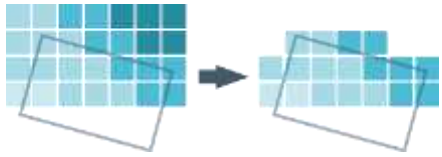
The RiskScape engine imports layers from project and data bookmark file settings, called during model execution. Data bookmark settings determine whether single or multiple layer files stored in a directory are imported as input data. These settings also specify geospatial, statistical, or other data manipulation operations to pre-process layer geometries or attributes. For instance, data bookmark settings for a comma-separated value (.csv) model input file will include a coordinate reference system and instructions on converting well-known text (WKT) into geometry objects, e.g. points, polylines or polygons.

2.2.3 Geoprocessing

The geoprocessing step transforms layer geometries for spatial sampling (see Section 2.2.4). Layers can represent various combinations of hazard, exposure, resource or area data layers as raster grids or vector geometries. Here, step-functions perform object geometry processing operations such as spatial and attribute filtering and logical functions, segmenting, joining, buffering, intersection and indexing (Table 3).

Table 3. Summary of model pipeline geoprocessing and spatial sampling step-functions.

Geoprocessing and Sampling Operations	Description	Graphical description
Attribute filter	Object-exposures representing single or multiple attributes in an exposure layer are identified from a conditional filter, creating a new exposure layer.	
Spatial filter by layer	Object-exposures that intersect single or multiple objects in an area layer are selected, creating a new exposure layer.	
Spatial filter by bounds	Object-exposures within the bounding box of a specified layer (e.g. hazard, resource or area) are clipped, creating a new exposure layer.	
Cut by segment	Geometries representing object-exposures are segmented by a specified distance either directly (e.g. polyline) or by grid (e.g. polygon).	
Cut by layer	Geometries representing object-exposures are segmented at points intersecting with object geometries or grids in another layer (e.g. hazard, resource, area)	
Buffer	Geometries representing object-exposures are enlarged by a specified distance. Point and polyline geometries are reclassified as polygons.	
Sample Centroid	The point location, polyline or polygon centroid is used to index the coverage data for spatial sampling.	
Sample Closest	The centroid of geometries representing the hazard object (e.g. grids, polygons) closest to the object-exposure is used to index the coverage data for spatial sampling.	

Sample All Intersections	All geometries representing hazard objects (e.g. grids, polygons) intersecting an object-exposure are used to index the coverage data for spatial sampling.	

Object geometries represented in layers determine the applicable geoprocessing operations. ‘Attribute filtering’ is performed on any layer that applies attribute specific conditional statements to extract objects. Object transformations are geometry specific. ‘Buffering’ operations apply to vector geometries representing exposures to expand an object’s area for spatial sampling (see Section 2.2.4). ‘Segment’ performs multiple operations that segment line or polygon object-exposures by either 1) a linear distance or area, or 2) the points where object-exposures intersect hazard object (i.e. original or buffered) boundaries. ‘Measure’ functions recalculate the new geometric dimensions of transformed line or polygon features. Geoprocessing operations are performed on object-exposures for single or multiple hazard or resource layers. The operations also facilitate spatial sampling of multiple hazard types and metric intensities at object-exposure locations.

2.2.4 Spatial sampling

Spatial sampling joins object attributes from hazard, resource and area layers to exposure layer objects. The sampling process creates a georelational coverage data file by converting non-exposure object attributes into indexed values at defined locations, e.g. centroid or vertices. Indexed values are extracted to the coverage data file from raster, vector or multi-dimensional data file formats (e.g. 2-dimensional, occurrence frequency, time-series etc.). Coverage data represents returned indexed values for any direct position within another layer’s spatial or spatio-temporal domain. Here, a step-function joins indexed values for non-exposure object attributes to object-exposures using lookup functions. This operation supports shared access to non-exposure object attributes at object-exposure locations for multiple purposes including, consequence analysis (see Section 2.2.5) and model output reporting (see Section 2.2.7).

Coverage data can represent single or multiple hazard types, metric intensities, and temporal occurrence at object-exposure locations. Spatial sampling operations based on object-exposure geometries determine descriptive statistics for indexed values representing hazard and other non-exposure object attributes (Table 3). Indexed values sampled at object-exposure centroids is the default operation while ‘all-intersections’ samples multiple index hazard intensity metric types and values (Table 3). Object-exposures buffered during geoprocessing expands the indexed value sampling area beyond original object geometries. The all-intersections operation derives descriptive statistics from all sampled indexed hazard intensity metric values within a specified proximity to object-exposures, e.g. flood inundation depths and velocities across a road surface.

2.2.5 Consequence analysis

The consequence analysis determines object-exposure impacts from single-hazard or multi-hazard events. Here, vulnerability functions determine the impact type and magnitude for specific object-exposures based on hazard types, metric intensities, or other phenomena. Vulnerability functions are independent from the RiskScape engine and implemented using Python programming language plugins. Python supports mathematical function libraries for vulnerability function (e.g. fragility functions, damage functions, damage-to-loss functions), implementation in continuous or discrete forms. For multi-hazard risk quantification, it is critical to support numerous functions that determine impact types and metrics for different hazard and exposure combinations (e.g. Silva et al. 2014; Tarbotton et al. 2015; Wilson et al. 2017; Huizinga et al. 2017).

The RiskScape engine executes vulnerability functions for impact calculation using modeller-defined classes. Classes are defined from logical expressions that identify conditions for function application to an object-exposure based on hazard and/or resource layer object attributes sampled at object-exposure locations (i.e. tuples). Python enables vulnerability function classes to be sequenced using conditional or nested statements. This facilitates object-exposure impact quantification for single or multi-hazard event occurrences. In multi-hazard risk analysis, the functionality organises vulnerability functions to calculate single or multiple impacts for object-exposures. Here, impacts from single (e.g. tsunami flow depth) or multiple hazard intensity metrics (e.g. tsunami flow depth, velocity and hydrodynamic force) caused by multiple simultaneous, sequenced or cascading hazard events (e.g. earthquake peak ground acceleration and liquefaction, followed by tsunami flow depth and velocity) escalate to reach an overall impact outcome for object-exposure (Kappes et al. 2012).

The consequence analysis produces an event impact table (EIT), used for subsequent risk analysis and model output reporting. The EIT combines structs, representing impact and model input data for each tuple, i.e. object-exposure. Descriptive statistics are calculated for impact metrics and reported from the EIT based on object-exposure attributes or geographical location (Section 2.2.7). Risk analysis then requires numerical aggregation of object-exposure impact metric values for hazard events (i.e. event ID) defined in model input metadata (Section 2.2.6).

2.2.6 Risk analysis

The risk analysis step quantifies the impact occurrence frequency for exposures based on hazard event probabilities. Here, modeller-defined step-functions execute risk metric quantification methods from EIT information. Impact metric values are numerically aggregated for EIT event IDs representing each hazard realisation of a stochastic hazard event set (SES). The paired event ID and impact value list facilitate deterministic and probabilistic calculations of common catastrophe risk metrics (e.g. Cardona et al. 2012; Velásquez et al. 2014; Goda and De Risi, 2017).

Deterministic Risk Analysis The RiskScape engine's plugin architecture enables modeller-customised step-functions for deterministic risk metric quantification. Object-exposure impact for a hazard event can be determined as follows:

$$Im_{ij} = f_C(H_{ij}|E_{ij}|R_{ij}) \tag{2}$$

where Im_{ij} is the impact to a tangible or intangible exposure (*E*) in response to the hazard and intensity (*H*) imparted by event *i* and resource (*R*) at location *j*. Vulnerability function impact values (f_C) are calculated for every event *i* affecting *E*. Single or multiple *H* can occur at location *j* from a defined event *i* set. The total impact values for object-exposures to event *i* can be numerically aggregated by:

$$Im_i = \sum_{j=1}^{N_E} Im_{ij} \tag{3}$$

where Im_i represents the total impact value for exposures and N_E is the total number of independent events *i*. Impact values can be aggregated and reported by object-exposure attribute (e.g. residential building) or geographic area (e.g. suburb) based on input data and geoprocessing step-function selections. A modeller-defined deterministic-based step-function can be applied to calculate the event expected impact (EEI) for each paired event ID and impact value in the EIT by:

$$\mathrm{EEI} = \frac{1}{N_E} \times \sum_{i=1}^{N_E} Im_i \tag{4}$$

Probabilistic Risk Analysis Exceedance probability impact (EPI) and average annual impact (AAI), are optional risk metrics calculated from modeller-defined step-functions. These risk metrics are generically termed to reflect the RiskScape engine's capacity to quantify $Im_i\ for$ tangible or intangible exposures. The EPI, often referred to as exceedance probability loss, is derived from independent variable Im_{ij} representing event i frequency as follows:

$$\text{EPI} = \sum_{i=1}^{N} Im_{ij} F(E_i) \tag{5}$$

where the probability of impact (Im_{ij}) is determine from E_i, and F is the annual frequency of occurrence for event i. N represents the total number of events i that are assumed to be independent. The AAI, often referred to as average annual loss (AAL) or expected annual damages (EAD), is calculated from EPI as follows:

$$\text{AAI} = \sum_{i=1}^{N_E} (EPI \cdot F) \tag{6}$$

where the event i impact is determined from the weighted average of the EPI impact for all independent events (N_E). Here, we describe several possible probabilistic-based step-functions to calculate EPI and AAI from the frequency of occurrence (e.g. AEP) for hazard events in the EIT (Table 4).

Hazard-based step-functions are applicable when the hazard event frequency of occurrence is pre-determined for each hazard event ID and impact value. Here, hazard event frequency (F) and impacts (Im_i) are paired for EIT event IDs (Table 4). Paired hazard event frequencies and impact values derive a hypothetical impact exceedance curve, often referred to as a loss exceedance curve or exceedance probability curve. EPI (Eq. 5) and AAI (Eq. 6) are calculated from represented event impact values using numerical integration or a probability distribution, assuming hazard event frequencies and impact values observe a monotonic relationship, i.e. as hazard event frequency decreases, impact values increase.

Table 4. Summary information for probabilistic-based step-function implementations in RiskScape model pipelines.

Risk Analysis Step-Function	Description	Example Case	Example EIT Configuration for Risk Metric Calculation

<table>
<tr>
<td>Hazard-based</td>
<td>A pipeline step-function to quantify risk metrics when hazard realisations in an SES have a pre-determined frequency of occurrence for a single hazard event.</td>
<td>An SES representing flood hazard events occurring in response to specific AEP discharge rates from a riverine source (e.g. Elmer et al. 2010).</td>
<td>
<table>
<tr><th>Event ID</th><th>AEP</th><th>Impact ($ M)</th></tr>
<tr><td>1</td><td>0.1</td><td>1</td></tr>
<tr><td>2</td><td>0.05</td><td>1.5</td></tr>
<tr><td>3</td><td>0.02</td><td>3</td></tr>
<tr><td>4</td><td>0.01</td><td>10</td></tr>
<tr><td>5</td><td>0.005</td><td>30</td></tr>
</table>
</td>
</tr>
<tr>
<td>Event-based</td>
<td>A pipeline step-function to quantify risk metrics when the frequency of occurrence for hazard realisations in an SES are assumed to be equal.</td>
<td>An SES representing all earthquake hazard events occurring at a geographic location over a 10,000-year (0.0001 AEP) time period (e.g. Cardona et al. 2008, 2012; Silva, 2018)</td>
<td>
<table>
<tr><th>Event ID</th><th>AEP</th><th>AEP*</th><th>Impact ($ M)</th></tr>
<tr><td>1</td><td>0.0001</td><td>0.0005</td><td>100</td></tr>
<tr><td>2</td><td>0.0001</td><td>0.0004</td><td>150</td></tr>
<tr><td>3</td><td>0.0001</td><td>0.0003</td><td>200</td></tr>
<tr><td>4</td><td>0.0001</td><td>0.0002</td><td>250</td></tr>
<tr><td>5</td><td>0.0001</td><td>0.0001</td><td>300</td></tr>
</table>
</td>
</tr>
<tr>
<td>Weighted event-based</td>
<td>A pipeline step-function to quantify risk metrics when numerous hazard realisations in an SES have a pre-determined frequency of occurrence for a single hazard event.</td>
<td>An SES representing the cumulation of volcanic ash fall simulations from multiple volcanoes where eruption AEPs may vary by several orders of magnitude (e.g. Jenkins et al. 2015)</td>
<td>
<table>
<tr><th>Event ID</th><th>AEP</th><th>AEP*</th><th>Impact ($ M)</th></tr>
<tr><td>1</td><td>0.01</td><td>0.058</td><td>100</td></tr>
<tr><td>2</td><td>0.005</td><td>0.049</td><td>150</td></tr>
<tr><td>3</td><td>0.02</td><td>0.044</td><td>200</td></tr>
<tr><td>4</td><td>0.005</td><td>0.025</td><td>250</td></tr>
<tr><td>5</td><td>0.02</td><td>0.02</td><td>300</td></tr>
</table>
</td>
</tr>
</table>

* Weighted mean of event AEP.

Event-based step-functions are applicable hazard event ID assumes an equal frequency of occurrence. Here, the number of SES hazard event realisations (e.g. > 1000) for a modeller-defined recurrence rate (e.g. 10,000 years or 0.0001 AEP) are weighted to determine a mean frequency of occurrence for each EIT event ID representing a hazard event realisation and impact value (Table 4). The event impact level (*EI*) frequency of exceedance (λ_{EI}) for a forecast time period (e.g. 1-year) is summed for all events exceeding the forecast *EI* (*EI* > *ei*) as follows:

$$\lambda_{EI} = \sum F_i(EI_i > ei) \tag{7}$$

The process is repeated for each defined *EI* within a modeller-defined recurrence rate. In addition, an impact exceedance curve determined from a representative probability distribution applied to each *EI* facilitates AAI (Eq. 6) calculation. For instance, assuming a Poisson distribution represents the SES hazard realisations as independent events, the probability of exceeding the *EI* ($P(EI_i > ei)$) in a forecast time period (T) is calculated as:

$$P(EI_i > ei) = 1 - exp(-\lambda_{EI} * T) \quad (8)$$

A *weighted event-based* step-function is applicable when the frequency of occurrence pre-determined for each hazard event ID and impact value. Here, the SES represents numerous hazard realisations for a single hazard event. For example, ashfall from a single volcano may cause 1000s of impact outcomes in response to ashfall deposition under changing wind or rainfall conditions; the range of eruption volumes needed to account for all potential impact outcomes can also range my several orders of magnitude. In such cases, the pre-determined frequency of occurrence (e.g. AEP) for each hazard realisation is used to weight and determine a mean frequency of occurrence (e.g. AEP) for each paired EIT event ID and impact value (Table 4). EPI (Eq. 5) is determined for each defined λ_{EI}, enabling AAI (Eq. 6) calculation from the impact exceedance curve for a representative probability distribution.

2.2.7 Model output data

Model pipelines executed in the RiskScape engine output data in tabular or common GIS file formats (e.g. .csv, .shp). These file formats enable data post-processing and analysis in open or proprietary GIS or spreadsheet software applications. In addition, web-compatible file formats (e.g. .kml, .json, .geojson) are options for model output data presentation on web applications, e.g. Google Earth. Model outputs (i.e. results) can be reported directly for object-exposures or filtered or grouped for numerical aggregation by attribute or geographic area. Risk metric derivatives generated by the RiskScape engine such as impact exceedance curves in graphical form, are a planned future reporting option to accompany the present reporting of tabular output data.

3 Discussion and conclusions

RiskScape is open-source software with a flexible modelling engine for multi-hazard risk analysis. The RiskScape engine implements a well-established framework for risk quantification. The engine is object-orientated and executes modeller-defined risk quantification workflows as model pipelines. Model pipeline steps and step-functions are linked by a domain-specific expression language. Step-functions analyse hazard, exposure, and vulnerability data across different spatio-temporal domains using geoprocessing and spatial sampling operations to transform data for impact calculation. The RiskScape engine supports deterministic and probabilistic risk quantification, with several probabilistic-based modes described in this paper. i.e. hazard-based, event-based, weighted event-based. Risk metric quantification methods are RiskScape engine independent, modeller implemented pipeline step-functions, configurable for different multi-hazard risk modelling contexts.

Risk modelling software is challenged by a requirement to implement risk quantification methods for multiple hazards and exposures across different spatio-temporal domains (Kappes et al. 2012). RiskScape complements a suite of application software addressing this challenge for single-hazard or multi-hazard risk analysis, e.g. HAZUS (Schneider and Schauer, 2006), CAPRA (Cardona et al. 2012) and OpenQuake (Silva et al. 2014), amongst others (Global Facility for Disaster Reduction and Recovery, 2015). RiskScape advances modelling software for multi-hazard risk analysis through several implementation features supporting modeller-defined risk quantification workflows. Firstly, the RiskScape engine is independent of both model input data and system prescribed data classifications. Impacts are calculated from modeller implemented scripts (via Python plugins), with logical expressions linking vulnerability functions to specific hazards, exposure types and attribute combinations. The functionality avoids the requirement for system defined vulnerability functions or curves that are limited to specific hazard and exposure attribute classifications. Secondly, geoprocessing operations spatially transform and align hazard and exposure object geometries. Geometry based geoprocessing operations (e.g. grid and polyline, polygon and grid, grid and grid etc.) represent complex hazard and exposure interactions. This functionality facilitates spatial sampling of single or multiple hazard types, metric intensities, and temporal occurrence information at object-exposure locations for impact calculation. These implementation features, supported by open geospatial consortium standard geospatial data files and operations, expedite RiskScape for modelling multi-hazard risk at any geographical location or scale.

Multi-hazard risk analyses are multi-part, posing technical issues for modelling spatio-temporal impact sequences in multi-hazard events. System prescribed data classifications and standardised model operations are common in modelling software to represent multiple hazard processes quantified as different metrics (Komendantova et al. 2014). However, this approach is limited to model prescribed impact outcomes. Here, the RiskScape engine supports modeller-defined workflows that analyse spatio-temporal impact sequences for multi-hazard events using two key model steps. Spatial sampling creates coverage data representing multi-hazard events coinciding or sequential at object-exposure locations. Python scripted conditional or nested statements establish a temporal multi-hazard impact sequence for object-exposures. Vulnerability functions are ordered by logical expressions to quantify escalating object-exposure impacts from the multi-hazard sequence. This functionality advances modelling software for multi-hazard risk analysis by enabling the escalating impacts from multiple hazard interactions to be quantify and aggregated to report an overall impact outcome. Impacts from multiple hazard interactions are not adequately represented when aggregating impact outcomes for independent single-hazard events (Kappes et al. 2012).

Several challenges focus RiskScape's continual development and improvement. The platform-independent RiskScape engine supports parallelised model pipeline steps and functions for efficient operation on a standard laptop. Model workflows must be scalable for analysing object-exposures, impacts, risk, uncertainties and sensitivities for single and multi-hazard events occurring at different spatio-temporal scales. Hazard events forming an SES may represent many hundreds of thousands of realisations, creating high computational demands. RiskScape's data processing engine supports multi-threading whereby, a greater number of processors or CPU cores are utilised to perform model pipeline steps and functions in parallel. Increasing computational resource efficiency supports scaling model pipelines to operate with increasingly large, complex spatio-temporal hazard and exposure model input data and reduces model runtimes for real- or near-real-time impact forecasting for single or multi-hazard events (e.g. tropical cyclones). Progressing model performance characteristics is critical for delivering timely information on hazard event risks and detailing model uncertainties and sensitivities.

RiskScape development in future will also focus on improving the user experience. The RiskScape engine CLI provides a light-weight tool for model pipeline configuration and execution. A CLI wizard assists modeller-defined pipeline configuration and execution however, a future planned web-based graphical user interface (GUI) for engine operation will support visual and graphical tools for model and data management. Software features that reduce barriers for modeller operation ensure modelling software such as RiskScape continue to meet the global demand for multi-hazard risk analysis tools.

4 Acknowledgements

The authors gratefully acknowledge RiskScape software funding contributions from the GNS Science (Contract No. C05X1702) and National Institute of Water and Atmospheric Research (Grant No. CARH2207) New Zealand Government's Strategic Science Investment Fund (SSIF), and the New Zealand Earthquake Commission (Contract No. 2019-MSA011).

Availability of data and materials

RiskScape is available at GitLab under the GNU GNU Affero General Public License license (GNU Operating System, 2007) for research use. The documentation is available online, including links to RiskScape use tutorials.

Conflict of interest

The authors declare no competing financial interests.

Supplementary Information 1: Glossary of key terms, abbreviations and acronyms

5 References

Aznar-Siguan G, Bresch DN (2019) CLIMADA v1: a global weather and climate risk assessment platform. Geosci Model Dev 12:3085–3097. https://doi.org/10.5194/gmd-12-3085-2019

Cardona OD, Ordaz M, Reinoso E, Yamín LE, Barbat AH (2012) CAPRA—comprehensive approach to probabilistic risk assessment: international initiative for risk management effectiveness. In: 15th world conference on earthquake engineering, Lisbon, Portugal.

Cardona OD, Ordaz M, Yamin, LE et al. (2008) Earthquake Loss Assessment for Integrated Disaster Risk Management. J Earth Eng 12:48–59. https://doi.org/10.1080/13632460802013495

Delmonaco G, Margottini C, Spizzichino D (2006) ARMONIA methodology for multi-risk assessment and the harmonisation of different natural risk maps. Deliverable 3.1.1, ARMONIA.

Elmer, F, Thieken, A. H, Pech, I, Kreibich, H. (2010) Influence of flood frequency on residential building losses. Nat Hazards Earth Syst Sci 10:2145–2159. https://doi.org/10.5194/nhess-10-2145-2010

GeoTools (2021) GeoTools The Open-source Java GIS Toolkit. https://www.geotools.org. Accessed 10 November 2021.

Global Facility for Disaster Reduction and Recovery (2015) Understanding Risk: Review of Open-source and Open Access Software Packages Available to Quantify Risk from Natural Hazards. World Bank Group, Washington, DC.

GNU Operating System (2007) GNU Affero General Public License, version 3, available at: https://www.gnu.org/licenses/agpl-3.0.en.html. Accessed 15 November 2021.

Goda, K., and De Risi, R. (2017) Probabilistic tsunami loss estimation methodology: stochastic earthquake scenario approach. Earthq Spectra 33, 1301–1323. https://doi.org/10.1193/012617eqs019m

Huizinga J, De Moel H, Szewczyk, W (2017) Global flood depth-damage functions: Methodology and the database with guidelines, EUR 28552 EN, Publications Office of the European Union, Luxembourg, 2017, ISBN 978-92-79-67781-6, JRC105688. https://publications.jrc.ec.europa.eu/repository/handle/JRC105688. Accessed 19 November 2021.

Jenkins, S.F., Wilson, T.M., Magill, C. et al. (2015) Volcanic ash fall hazard and risk. Global volcanic hazards and risk, pp173–222.

Kappes MS, Keiler M, von Elverfeldt K et al (2014) Challenges of analyzing multi-hazard risk: a review. Nat Hazards 64:1925–1958. https://doi.org/10.1007/s11069-012-0294-2

Komendantova N, Mrzyglocki R, Mignan A et al (2014) Multi-hazard and multi-risk decision-support tools as a part of participatory risk governance: feedback from civil protection stakeholders. Int J Disast Risk Res 8:50–67. https://doi.org/10.1016/j.ijdrr.2013.12.006

Paulik R, Stephens SA, Wild A et al (2021) Cumulative building exposure to extreme sea level flooding in coastal urban areas. Int J Disast Risk Res 66:102612. https://doi.org/10.1016/j.ijdrr.2021.102612

Rossi L, Rudari R, et al (2016) RASOR Project: rapid analysis and spatialisation of risk, from hazard to risk using EOdata, Geophy Res Abs Vol. 18, EGU2016-15073, p. 15703

Scawthorn C, Flores P, Blais N, et al (2006) HAZUS-MH Flood loss estimation methodology. II. Damage and loss assessment. Nat Hazards Rev 7:72–81. https://doi.org/10.1061/(ASCE)1527-6988(2006)7:2(60)

Schneider PJ, Schauer, BA (2006) HAZUS - Its Development and Its Future. Nat Hazards Rev 7:40–44. https://doi.org/10.1061/(ASCE)1527-6988(2006)7:2(40)

Silva, V (2018) Critical Issues on Probabilistic Earthquake Loss Assessment. J Earth Eng 22:1683-1709. https://doi.org/10.1080/13632469.2017.1297264

Silva V, Crowley H, Pagani M, et al. (2014) Development of the OpenQuake engine, the Global Earthquake Model's open-source software for seismic risk assessment. Nat Hazards 72:1409–1427. https://doi.org/10.1007/s11069-013-0618-x

Tarbotton C, Dall'Osso F, Dominey-Howes D et al. (2015) The use of empirical vulnerability functions to assess the response of buildings to tsunami impact: Comparative review and summary of best practice. Earth Sci Rev 142:120–134. https://doi.org/10.1016/j.earscirev.2015.01.002

United Nations International Strategy for Disaster Reduction (2015) Sendai Framework for Disaster Risk Reduction 2015–2030, UNISDR, Geneva.

Williams S, Paulik R, Weaving R et al. (2021) Multiscale Quantification of Tsunami Hazard Exposure in a Pacific Small Island Developing State: The Case of Samoa. GeoHazards 2:63-79. https://doi.org/10.3390/geohazards2020004

Wilson G, Wilson TM, Deligne NI et al. (2017) Framework for developing volcanic fragility and vulnerability functions for critical infrastructure. J Appl Volcanol. https://doi.org/10.1186/s13617-017-0065-6

Woods RJ, Thomas, KL, Scheele F, Canessa S, Horspool NA (2021) Queenstown Debris Flow and Rockfall Loss Modelling for Land-Use Planning Policy Options. GNS Science CR2021/07: 35p.

Velásquez CA, Cardona OD, Mora MG et al. (2014) Hybrid loss exceedance curve (HLEC) for disaster risk assessment. Nat Hazards 72:455–479. https://doi.org/10.1007/s11069-013-1017-z

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryInformation1.docx