**Risk MAP CDS**

**OpenHazus**

**Whitepaper**

Prepared by the IBM Risk MAP CDS Team

NT

For:

Department of Homeland Security

Federal Emergency Management Agency

Mitigation Division Washington, D.C.

Hazus® is a trademark of the Federal Emergency Management Agency

**Revision History**

| Version # | Version Date | Summary of Changes | Team/Author |
| --- | --- | --- | --- |
| 0.1 | 1/3/2019 | Initial Version | CDS Risk MAP Team |
| 0.5 | 1/12/19 | Rewrite of purpose | CDS Risk MAP Team |
|  |  |  |  |
|  |  |  |  |

**Contents**

Executive Summary

I Introduction

1 Background

2 Purpose

3 Scope

II The OpenHazus Initiative

1. Assumptions and Dependencies

III Current State of Hazus

1. User Descriptions and Key Features
2. Levels of Analysis
   1. Analysis Based on Default Information
   2. Analysis with User-Supplied Inventory
   3. Analysis with Advanced Data
   4. Anticipated breakdown of OpenHazus users
3. Typical Workflows
   1. Flood model workflows
   2. Hurricane model workflows
   3. Earthquake model workflows
   4. Tsunami model workflows

IV Solution Framework

1. Modularization
2. Independence from Desktop GIS
3. Advantages of the Open Character
   1. Training and outreach benefits
4. Inventories as an Indispensable Aspect of Open
   1. Requirements for demographic inventory data
   2. Requirements for building and critical infrastructure data
   3. Requirements for depth grids
   4. Requirements for the results database

V Architecture

1 Hazard

2 Analysis

3 Mapping of modules to user types

VI Product Review

1 Description of Open Source Desktop GIS

1. GRASS
2. QGIS
3. uDig
4. gvSIG
5. SAGA
6. Open JUMP

2 Description of Open Source Web Mapping Applications

1. MapServer
2. MapGuide
3. OpenLayers
4. degree
5. GeoMoose
6. MapBender
7. GeoServer
8. Carto
9. PyWPS

3 Description of Open Source Geospatial Libraries

1. GeoTools
2. GDAL/OGR
3. GEOS
4. GeoTrellis
5. GeoWave
6. STAC

4 Description of Other Open Source Platforms

1. R
2. Repast Simphony
3. Eclipse
4. Jupyter

5 Decision Analysis Report

VII Database

1 PostGIS

2 SpatiaLite

3 rasdaman

4 OLAP Databases

5 Non-relational Databases

6 Decision Analysis Report

VIII Analysis of Alternatives

1 Preferred Solution

2 Security Considerations

3 Assumptions and Dependencies

IX Implementation

1 Inventory Access

2 Modelling access

X Testing and Prototyping

1 Optimize extract by value

2 Microsoft Building Footprints as OpenHazus Foundational Inventory

3 Importing from the MIP, feasibility

4 Inventory coverage of flood products

5 Open Source Utility for Hazus Data Export and Reporting

6 Streamlined Nationwide Inventory Updates using HIFLD

7 Open Source, Web-Based Hazus Flood Module

A1 Definitions, Acronyms, and Abbreviations

A2 References

**Executive Summary**

The mission of Hazus is to combine science and technology to support FEMA’s mission of “…leading and supporting the Nation in a risk-based, comprehensive emergency management system of preparedness, protection, response, recovery, and mitigation.” Hazus is maintained and operated with the Risk Mapping, Assessment, and Planning (Risk MAP) program and supports its vision to “…deliver quality data that increases public awareness and leads to action that reduces risk to life and property.”

**I. Introduction**

The purpose of this whitepaper is to define the functionality, interfaces, performance, attributes, and design constraints of OpenHazus the proposed Open Source version of FEMA’s Hazus risk analysis software. It is based on a Product Requirements Assessment (PRA) focused on the major capabilities and features needed by Hazus stakeholders. This document will not define additional capabilities but focuses on the provision of the capabilities defined in the PRA. The hypothetical end product of this effort has been coined “OpenHazus” to denote not only a shift to Open Source software and technology, but also a shift toward datasets, methodologies, and model functionalities that are more accessible and transparent for Hazus stakeholders. This whitepaper includes sections addressing architecture, product research, and prototyping. Because OpenHazus will be distributed for free by the federal government, other topics, such as pricing, competition analysis, marketing issues, are not relevant and are not addressed.

**I.1 Background**

Hazus is a software suite within the risk mapping, assessment and planning section (Risk MAP) of the Federal Insurance and Mitigation Administration (FIMA) of the Federal Emergency Management Agency (FEMA), which is one of over 20 top-level units of the Department of Homeland Security (DHS). FEMA started Hazus in 1992 in response to the National Academy of Sciences finding that no nationally consistent natural hazards loss estimation methodology was available that met the needs of the emergency management community for quantitative risk assessment. The Hazus earthquake risk assessment methodology and modelling capability was established in 1997 with the help of the National Institute of Building Sciences utilizing GIS-based software applications. Hazus exclusively modelled earthquakes until 2004, when it expanded to include flood and wind loss estimation models. Hazus subject matter experts from leading scientific and engineering practitioner and academic communities continue to improve the analytical accuracy and software speed of Hazus models and expand model functionality and community involvement for Hazus stakeholders.

Hazus provides a cost-effective method of identifying and addressing the impacts associated with a natural disaster. Hazus results allow members of the emergency management community to develop policies for preparation, protection, response, recovery, and mitigation of natural hazards that aim to decrease the risk of future loss. Hazus is also used in support of the Disaster Mitigation Act (DMA) and to develop Hazard Mitigation plans to provide critical information for decision makers. When combined with detailed analyses and expert knowledge, the default datasets and model parameters included with Hazus provide an effective risk assessment tool. The overall objective of the existing Hazus project is to implement a nationally applicable set of standardized multi-hazard methodologies for estimating potential wind, flood, and earthquake losses on a regional scale. Hazus results are intended for use by local, state, and regional officials for risk assessment, planning and stimulating mitigation efforts to reduce losses from hurricanes, severe floods, and earthquakes and preparing for emergency response and recovery following these events. Local, state, and federal authorities and risk reduction policies are driving the need for the Hazus software package as specified in NEHRP, NHRAP, and NWIRP ([Reference needed](https://www.nist.gov/el/...and.../national-windstorm-impact-reduction-program-nwirp)) and exemplified by the need to provide risk assessments for state-owned infrastructure. Depending on the capability built in for each hazard, Hazus is also used to produce rapid damage estimates immediately following an event. Hazus has shown to be capable of integrated multi-hazard loss estimation with the following major features:

* Models for estimating damage from earthquakes, floods, hurricanes and tsunamis
* Capability to run both deterministic and probabilistic scenarios
* A single, integrated set of functions for study region creation for all three models
* Geographic Information System (GIS) functions
* Capability to receive user-supplied input for all three models to generate more refined loss estimations
* Varying degrees of real-time analysis for each hazard
* State-of-the-art software, fully documented with metadata for all databases

Hazus is distributed free of charge. However, it is designed to run as an extension to the commercial GIS software ArcGIS Desktop, developed and distributed by ESRI, Inc. This software costs approximately $6,500 per seat, which is cost-prohibitive for many potential users. ESRI is phasing out its development of ArcGIS Desktop in favor of a more modern architecture called ArcGIS Pro. Hazus is so tightly integrated with the dynamic link libraries of ArcGIS Desktop that a switch to ArcGIS Pro would require a complete rewrite of Hazus. While there are other reasons to pursue “open” solutions (see section 1.2), the significant impending cost of rewriting Hazus to maintain compatibility with ArcGIS Pro has precipitated the need to redesign Hazus to be independent of external GIS viewers and provided the opportunity to pursue a new set of requirements in Chapter III.

**I.2 Purpose**

The design and functionality of Hazus have become outdated as interest in risk information has widened and the technical landscape for data analysis has dramatically shifted. Members of the Hazus community have been discussing ways to redesign Hazus in order to effectively broaden its scope and support its role as a leading publicly available risk modeling platform.

In an age of increasingly frequent and extreme weather events, applied researchers in a range of social and natural sciences are now dealing with the complexities of long-term planning, protective infrastructure, public health, and social vulnerability. These new applications have widened the scope of academic communities interested in working with Hazus and, in turn, increased the amount of expert input available for the continued improvement of Hazus modelling capabilities. Demand for quantitative risk information has increased alongside this widening user base, evident in new regulatory requirements and incentives offered by FEMA programs for communities or projects that implement data-driven decision making.

As the audience and demand for quantitative risk information have grown, the technical infrastructure available to derive such information has drastically changed. Cloud computing capabilities have become standard, data visualization tools are numerous and designed for non-technical users, and traditional GIS tasks are increasingly integrated into generalized analytical workflows with fewer distinctions between spatial and non-spatial processes. In fact, the functionality of Hazus and other risk modeling technologies do not necessitate a GIS at all. Hazus dependency on ESRI software was initially an advantage for Hazus because the majority of technically well-versed users had GIS experience and ESRI’s ArcGIS Desktop platform has been the most widely used GIS in the United States for many years. However, as the community of stakeholders working with spatial data spreads beyond traditional GIS users, the need arises for visualization and analysis tools that effectively integrate spatial and non-spatial data in systems that are agile, accessible, and transparent – regardless of their interface.

A 2018 inventory found that out of XXX lines of code, Hazus only employs XXX lines of geospatial analysis. And while maps are a highly efficient tool for communicating natural hazard risk information, a large fraction of Hazus model results are more effectively communicated through traditional visualization techniques or their modern interactive counterparts (charts, graphs, infographics, etc.) The tight coupling of Hazus with ArcGIS Desktop places a significant burden on users who want to customize the software and extend its capabilities and on the Hazus Program itself, whose resources are largely spent on cumbersome software releases that employ complex code changes to address otherwise straightforward functionality upgrades. If GIS capabilities represent a small and burdensome fraction of Hazus functionality, it becomes practical to leverage available Open Source geospatial libraries to accomplish spatial data visualization and analysis, rather than designing all Hazus capabilities around a GIS interface.

Similarly, a relatively small percentage of Hazus users require a desktop environment for their risk modeling projects. Users with a legal obligation to maintain secure risk modeling data inputs (actuaries, certain military planners) represent an estimated 10% of Hazus use cases. Given the ongoing shift in computing technologies away from local environments and onto cloud resources, leveraging a web-based architecture for Hazus could fulfill 95% of user needs while increasing modeling speed and access and radically expanding the transparency of model methodologies, data, and results.

Open Hazus has been conceptualized as a toolbox of multi-hazard methodologies that may but do not have to use GIS to visualize the spatial distribution of risk to given assets. Removing the dependency not only on ESRI GIS software but also on traditional GIS architecture in general can significantly broaden the range of potential Open Hazus users. Hazus users and developers would benefit from two different aspects of the proposed “toolbox” of open software architecture. In a highly modularized system leveraging as-needed open source analytical libraries, each tool is well documented and can be accessed individually via an Application Programmer’s Interface or API. Users can employ only those Hazus modules that make sense for their projects, and developers can more efficiently expand and update Hazus functionality by accessing separate analytical modules rather than editing a large, rigid and interdependent workflow. The other beneficial aspect of an open architecture is the reliance on free and open source software (FOSS), which would make it even easier to customize Hazus for many uses beyond its original scope. As advanced users develop their own customized versions of Hazus analytical processes, the Hazus Program can devote more resources toward collaborating with these users to review new model versions and integrate them into the official Hazus architecture, thereby leveraging the largely untapped expertise of the nationwide risk modeling community. A transition to FOSS would also increase the much-requested transparency of data and methods relied upon by federal agencies, allowing Open Hazus users to access key aspects of risk model architecture, thereby increasing trust in model outcomes.

**I.3 Scope**

The Purpose described above has driven the development of new baseline requirements for OpenHazus as a leading publicly available risk modeling toolkit. The Hazus Team has developed 10 priority Requirements for OpenHazus:

① Loss estimation tool. This is the core application of OpenHazus that takes precedence over other potential demands in supporting the mission of FEMA and of the Risk MAP program in particular. Any rewrite of Hazus will have to fulfill its current set of capabilities related to the estimation of social (fatalities, displacement and injuries) and building (damages and economic loss) impacts due to natural hazards. This requirement does not include estimating the extent and magnitude of potential hazards. OpenHazus implementation will emphasize the use of external, authoritative hazard datasets..

② Data/results-sharing platform. In order to better meet the needs of non-technical users interested in risk model results, OpenHazus will include an online platform for viewing, sharing and downloading generalized but authoritative risk assessment results from across the U.S. This includes a standardized way of storing and accessing metadata associated with available risk assessment results, including the lineage of how the data were created – even if this happened outside of OpenHazus. The standards requirements for metadata, as defined by FGDC and OGC, must be adhered to. A first expansion of these capabilities includes a seamless link between OpenHazus and FEMA’s MIP and MSC to further integrate FEMA’s risk assessment and mitigation programs.

③ Support import and integration of authoritative hazard data. In the spirit of openness, as outlined in section I.2, OpenHazus must be able to integrate authoritative data from providers, including the USGS, NOAA, as well as FEMA’s own flood hazard engineering data. Parallel to that, stakeholders should be able to upload user-defined hazard data with standardized QA/QC measures to a sandbox, where it can be shared more broadly through a public domain repository with FEMA’s stamp of approval after it has passed internal quality control.

④ Modularization. One of the advantages of a thorough overhaul of the Hazus software architecture is the possibility to create a library of modules that encapsulate individual processing steps. Basic users can choose to interact with only those analytical steps that apply to their project goals, rather than adhering to a fixed workflow (though user-friendly workflow suggestions will be provided.) Experienced users and developers should then be able to create their own workflows, adapting them to agency and application needs.

⑤ User-friendly GUI. The current user interface is an extension of GIS desktop software. With its separation from GIS, the OpenHazus user experience can be improved significantly. Whereas Hazus was always conceived as a desktop (workstation)software, OpenHazus should be flexible enough to serve stakeholders in a web-based, desktop-based, and even mobile environment. Not all functions will be available in all environments, but the transition should be seamless from a UI/UX perspective.

⑥ Secure/private data integration. An estimated ten percent of Hazus stakeholders are working with PCII and PII data that require secure storage, access and communication mechanisms. Conversely, the vast majority of Open Hazus stakeholders will be better served by web services. As Open Hazus is built around the latter, the needs for secure or private data integration on desktop and mobile computers must be addressed without impacting the UI/UX design of the majority.

⑦ Improve Hazus-provided inventory data. A significant obstacle to a more widespread use of Hazus is the amount of effort it takes to create inventory data. In recognition of this impediment, FEMA has, in cooperation with DHS’s HIFLD support team, developed an incomparable resource of some 16 GB of generalized inventory data that are editable through the custom-designed Comprehensive Data Management System (CDMS). While these data could be a good starting point for authorities who do not have the personnel resources to develop their own inventories, they are provided only as backend SQL tables during routine Hazus installation and are not accessible to average users for custom analysis or visualization. Furthermore, demands for accurate risk information and targeted risk reduction have outgrown census tract-level analyses and with the availability of building-level data, expectations have grown for CDS to provide a centralized national building data set. As specified in requirements ② and ③, in which users must have access to risk model results and authoritative hazard data inputs, establishing OpenHazus in a web service environment should facilitate users uploading their own improved inventory data into access-limited sub-clouds. As with the flood data in ③, a process should be established that allows to merge higher quality inventories with the main repository.

⑧ User must not pay to use Hazus. OpenHazus as a methodology toolset should remain free, as should access to core national datasets. Private for-profit entities may be charged a fee for usage that exceeds five percent of the OpenHazus server load.

⑨ Must be thin client. For most stakeholders, i.e., those who are not working with private or secure data, most of the data and all of the processing should occur on a (web) server. One of the main issues with Hazus is that software updates must be synchronized with updates for the operating system and dependent GIS software, and often these updates conflict with other locally installed software. Many Hazus installations are therefore isolated from the rest of the enterprise workflow, requiring significant hardware and software resources and maintenance and impacting the overall security of installation. A thin client consists of a relatively small application that can be installed on a variety of platforms such as MS Windows, Mac OS, Linux, Chrome, iOS, Android, etc.

⑩ Ability to automate workflows. A number of model runs can take quite a while to execute, especially if users want to run an ensemble of scenarios. It is therefore desirable to have the ability to run complete workflows in batch mode without any user interaction.

**II. The OpenHazus Initiative**

The Hazus Strategic Plan (FY17-FY21) supports Hazus and Risk MAP missions by identifying key goals and the supporting objectives required to achieve these goals. Hazus provides stakeholders with a dependable view of the potential risks they face and thereby encourages hazard mitigation. Primary stakeholders include decision makers, such as state and local elected and appointed officials, who provide the leadership and resources to implement Hazus as well as practitioners like state and local emergency managers, planners, and floodplain managers, who are directly responsible for mitigation planning and other emergency management tasks. As indicated in Table 2.1, the 2017 Strategic Plan identifies five goals, three of which are directly addressed by this whitepaper, while the other two represent a necessary background for successful implementation.

Open Hazus will have a single online resource that will allow Hazus users easier and quicker access to relevant information. This online portal will make basic results discoverable and available for the appropriate model with updates scheduled at a set frequency, thereby considerably widening the user base of Hazus. Standardized results will allow complicated technical model outputs to be easily digestible and better understood by the end user. This will also help bridge the gap between risk analysts and decision makers by making the results more immediately accessible by leadership.

Open Hazus will have an easily searchable online repository for users to share inventory including User Defined Facilities (UDF), best practices, and results. This requires the development of a quality control process and committee to vet new data in collaboration with Production and Technical Services (PTS) Contractors, Cooperating Technical Partners (CTP), and FEMA Community Engagement and Risk Communication (CERC) projects.

A truly OpenHazus will make it dramatically easier to solicit contributions from subject matter experts because they can now comment and improve upon methodologies that are available at source code or pseudo source code level. This will build the end user’s confidence in Hazus results and increase the software’s accuracy. One of the main advantages of Open Hazus will be the establishment of a community of users and developers that, like in many FOSS projects, handle a large share of helpdesk duties and act as a conduit between actual users and developers. The increased transparency and accountability should improve stakeholder confidence significantly.

*Table 2.1*

|  |  |  |
| --- | --- | --- |
| ***Goal*** | ***Objective*** | ***Open Hazus*** |
| Enable Hazus access and usability for a broader group of users | * Consolidate Hazus online resources into a single portal | 🗹 |
| * Standardize and simplify Hazus results | 🗹 |
| * Provide nationwide basic results for earthquake, flood, and hurricane online | 🗹 |
| * Create and maintain online repository of Hazus results and products driven by user submitted content | 🗹 |
| Ensure Hazus software is reliable, scalable, and up-to-date | * Increase Hazus stability | 🗹 |
| * Integrate subject matter experts in the development process to maintain modeling accuracy |  |
| Continue to update methodologies with the latest established science | * Perform routine assessment of methodologies to ensure they reflect the latest science |  |
| * Perform post-event studies comparing Hazus modeling against real world results to assess methodology accuracy |  |
| * Create national resource for developing standardized risk assessments | 🗹 |
| Further engage the community | * Ensure training is robust, applicable, and timely leveraging real-world scenarios |  |
| * Perform routine stakeholder analysis | 🗹 |
| * Increase transparency by communicating the development and release schedules, modeling changes, and other Hazus updates | 🗹 |
| Establish a structure for the updating, maintenance, and implementation of all Hazus elements | * Develop performance metrics |  |
| * Draft a data management plan |  |
| * Coordinate supporting documentation and training materials with software releases | 🗹 |
|  |

**II.1 Assumptions and Dependencies**

Requirements of the OpenHazus whitepaper development are affected by the following:

* Project funding and schedule
* Changes in project requirements
* Availability of recent, high-resolution demographic and inventory data (e.g. U.S. Census)
* Availability of required features in the Hazus application shell
* Availability of building stock data, damage functions, and loss modelling functions required to perform the loss estimations
* Availability of external sources of high-quality authoritative hazard data
* Operation within a common RDBMS

Key assumptions behind this Open Hazus whitepaper are described below.

Proposed Hazus software architecture supports the requirements outlined above.

* There is a common interface to items such as the inventory, hazard, analysis and results menus, study region builder, as well as inventory preparation tools and results exporting.
* The Risk MAP CDS Team is responsible for system integration and shall coordinate and support all tasks associated with integrating the hazard models with the overall OpenHazus software.

The following requirements are considered to be desirable but will be pursued after development of the initial versions of OpenHazus:

* Allowing users to define their own custom building types.
* Revising damage functions.
* Optimizing average annualized loss.
* Facilitating ESRI development of Hazus plug-in on newer versions of ArcGIS as well as Windows Operating Systems.
* Optimizing Fire Following Earthquake.
* Data updates when available.

**III. The Current State of Hazus**

This section describes

**Hurricane Workshops**

**Flood Workshops**

**Tsunami Workshops**

**Disaster Operations Workshops**

**Basic Hazus Workshops**

**CDMS Workshops**

**III.1 User Descriptions and Key Features**

Hazus is designed for use by federal, state, and local emergency planners, and hazard mitigation officers. These planners will use the tool for both long-range and short-term emergency response planning.

This section describes intended users of the Hazus model and their operating environment. These profiles are based primarily on interviews conducted with potential users and input received from a user groups that were established to work with the software development team.

At the local level, the Hazus model is used to address land use management issues such as floodplain management, buy-out programs, and building code development and enforcement. As users replace default data with accurate locally available data, additional applications, such as performing benefit/cost analyses for proposed mitigation measures, will emerge. It is anticipated that users at the local, state and federal levels will use the program to meet their programmatic needs and policy decisions.

Each user profile is discussed below.

**III.1.1 Local Floodplain Coordinators/Managers**

The majority of potential users of the Hazus flood model are local floodplain coordinators at the city, township, or county levels. These users are interested in identifying areas in their jurisdictions that are subject to flooding for a variety of return periods and the inventory (all types) that are subject to that inundation. Specific application requirements will vary, depending on the size and technical sophistication of the jurisdiction, but user interviews conducted as part of this project indicate these to be key foreseen applications.

For example, local users are interested in identifying flood-prone areas, assessing the implications of existing land uses, and evaluating the effectiveness of potential mitigation measures to prevent future losses. Some mitigation measures mentioned include development regulations, acquisition and relocation programs, and elevating vulnerable structures.

These users are also interested in the ability to effectively communicate flood risk issues to the general public and disseminate information about floodplain management and the existing threat to the community.

The general conclusion drawn from the interviews was that local users will likely replace at least some default data with their own. Most claimed to have better elevation models and building inventories and are interested in a having a portfolio capability for their repetitive loss structures.

**III.1.2 State Users**

At the state level, likely users will be state floodplain managers and hazard mitigation officers. These users are more interested in performing regional flood vulnerability assessments of watersheds in states where repetitive losses are a continuing problem. These users are also likely to be interested in conducting benefit/cost analyses of proposed building codes, land uses and zoning regulations to reduce the impact of losses to their states. Because state users are likely to perform the analysis on larger areas, they will be less likely to replace the default data because of the cost involved in data collection, although a possible exception may be elevation and essential facility data. State users may also perform the Hazus analysis for local users who are unable to perform their own analysis (most likely small communities or unincorporated areas).

**III.1.3 Federal Users**

Federal user requirements vary greatly from agency to agency. Currently, the project team has identified the Federal Emergency Management Agency (FEMA), the U.S. Army Core of Engineers (USACE) and the U.S. Geological Survey (USGS) as the primary federal users.

FEMA has identified the need to develop national loss estimates to meet its obligations to Congress. Developing a loss estimation tool that uses a consistent national baseline inventory and methodology will solve this problem. Using national baseline data supplemented with other flood-related studies should ensure a continual and consistent capability for FEMA.

USACE has identified the need for a software model with a national baseline building inventory that would satisfy the need for preliminary loss estimation associated with flood projects (structural and non-structural). USACE utilizes the Hazus national baseline inventories and has expressed interest in the watershed capabilities of the Hazus flood model.

Other potential federal users include the U.S. Small Business Administration, U.S. Bureau of Land Management, the U.S. Fish and Wildlife Service, and the U.S. Department of Agriculture. No effort to establish requirements for them has been made because their level of usage is expected to be minimal.

**III.1.4 Special District Users**

Special districts, including flood control districts and stormwater services, are part of the Hazus Flood User Group. Their requirements have paralleled other users at the local and regional levels. In addition, Regional Planning Commissions frequently use Hazus for multi-jurisdictional mitigation planning.

**III.1.5 Consultants and Private Sector Users**

A special set of users include the consultant and private sector users who perform project analysis for local, state and federal clients or are supporting other organizations and other private clients who have requirements to quantify potential losses. The cottage industry for loss estimation has expanded following the Hazus software releases. Specific application requirements vary greatly depending on the goals of the client, and reflect the geographic area of responsibility of the client.

The private sector users may fall into the following classifications:

* Consultants hired by government agencies to define potential losses for the development of mitigation programs or other policy-related decisions. This includes consultants assisting agencies in examining the impact of land use planning decisions including floodplain management.
* Companies investigating their exposure to local hazards. This includes national companies that use the model to examine their facilities over the entire U.S.
* Risk managers assisting national or local companies in their Business Continuity Planning.
* Corporations assisting FEMA with multi-hazard risk analysis including the quantification of risk and risk communication.

This list is not meant to be an exhaustive examination of potential users within the private sector but should give some idea of potential applications.

**III.1.6 Levels of Expertise**

Users can be broken into two groups: those who are performing a study and those who are using the results of the study. For some studies these two groups will consist of the same people, but generally this will not be the case. However, the more interaction that occurs between these two groups, the more accurate and useful the study will be. End users of a loss estimation study need to be involved from the beginning to make results more usable. Those who are performing the study may consist of multi-disciplinary teams and must, at minimum, have a basic understanding of applicable hazards and their consequences. In many cases, the results will be presented to audiences (i.e., city councils and other governing bodies) that have little technical knowledge of the hazard loss process. However, their participation throughout the process helps to ensure the results are appropriate for the decisions they are required to make.

It is assumed that a loss study will be performed by a multi-disciplinary team consisting of severe storm/earthquake/tsunami experts, structural engineers or architects, economists, sociologists, emergency planners, GIS professionals and representatives from the group who will be reviewing/using the loss estimates. These individuals are needed to develop hazard scenarios, develop and classify building inventories, provide and interpret economic data, provide information about the local population, and provide input as to what types of loss estimates are needed to fulfill the goals of the loss study. It is important to have an interdisciplinary team as, for example, hurricanes frequently also produce coastal and/or inland flooding or local tsunamis occur in combination with earthquake ground-motion and deformation.

It should be noted that the involvement of the ultimate users of the study on the team is very important. End users of the loss estimation study (i.e., decision makers) need to be involved from the beginning to make results more usable. If a local or state agency is performing the study, some of the requisite expertise can be found in-house. Experts are generally found in several departments: building permits, public works, planning, public health, engineering, seismic and tsunami science, information technologies, finance, historical preservation, natural resources, and land records. Although internal expertise may be most readily available, participation of individuals from academic institutions, citizen organizations, and private industry cannot be underestimated.

Although a loss study can be performed with a minimum of expertise using only the defaults and national baseline inventory provided by the computer program, the results of such a study should be interpreted with caution and applied appropriately, as default values have a great deal of uncertainty associated with them. If the loss estimation team does not include individuals with expertise in the areas described above, then it is likely that one or more outside consultants may be required.

Unless scenarios have already been developed and documented for the study region, the user may require the expertise of a seismologist, hydrologist, meteorologist or wind engineer when defining deterministic scenarios. Even if a scenario event has been documented, it may be defined using parameters or a format that are different than those used in Hazus. In this case, an expert will be needed to review the scenario and describe it or translate the input into one of the formats supported by the hazard model. A scenario event that is defined without an in-depth understanding of events affecting the region may not be appropriate for the loss study. For example, a user could define or integrate an extreme hazard event that is not reasonable for the study area.

If the user intends to modify the default data or parameters, it is likely that s/he will need input from someone with expertise in the field. For example, if the user wishes to change default percentages of model building types for the region, s/he will need the input of a structural engineer who has knowledge of design and construction practices of the region. Modifications to defaults in the economic loss models will require input from an economist.

**III.1.7 Key User Features**

During development of the Hazus methodology, it became clear that user input would significantly increase acceptance of loss model methods and results. Interviews of users covering all levels and functions identified key features that would facilitate the user’s job and help identify how they would define success in applying the results. The sections below provide an overview of key points identified by users; however, no products currently exist that completely fulfil these requirements.

*Identification of Flood-prone Areas*: Many floodplain managers stated that identifying areas subject to flooding is a key concern. Most felt their existing Flood Insurance Rate Maps (FIRMs) (including DFIRMs) were out of date or did not reflect the entire flood hazard for their community, including alluvial and pluvial flooding, or recent development that leads to increased urban runoff and subsequent increases in flood elevations.

Floodplain managers want a tool that uses a consistent methodology to allow them to compare new flood studies with existing studies to project changes or areas of concern. They also want a tool to allow them to perform “what if?” analyses to determine the impact of potential mitigation scenarios or increased water volumes above the 100-year or design flood levels.

*Prioritization of Repetitive Loss Structures for Future Mitigation Projects*: A high priority of both FEMA and the NFIP is to identify structures that have suffered repetitive losses. Because this responsibility will likely fall on local officials, interviewed users ranked this capability very high in importance. Currently, no comprehensive tool exists that allows users to conduct a spatial analysis of the cause of these losses and to identify and prioritize cost-effective mitigation measures. A tool that can provide this capability is deemed highly desirable by interviewed users.

*Building Standards and Floodplain Management Regulations*: Associated with the prioritization of repetitive loss structures is the requirement to manage existing and future development within and near the floodplain. Closely associated with proper land use planning, developing building standards and regulations to control and restrict development in and near floodplains is a key concern.

Providing a methodology to identify the effects of standards and regulations is a key benefit. Most users feel that a scientific approach would help them approach local elected officials when requesting approval of standards and regulations. Additionally, users feel that tools such as Hazus provide officials with supporting information to assist in their effective decision making.

*Flood Warning*: While this requirement is self-explanatory, no tools currently exist that can be used in real time or near real time to identify the potential impacts for projected flood elevations. Users want a tool that can help the emergency management community project areas of inundation and provide warning so that residents can perform flood prevention/proofing, remove or move contents, and evacuate as necessary.

*Flood Loss Estimation Analysis and Benefit/Cost Analysis*: Tools to estimate flooding (such as USACE-HEC) are complex and require a minimum level of technical expertise to use. The application of such tools will assist in assessing benefit/cost of specific flood control projects.

Users have identified the need for a tool to conduct a lower-level analysis that can assist them in rapidly demonstrating the benefit/cost of various project mitigation measures in order to prioritize them. It is important that the tool produce loss estimates at resolutions smaller than census tract (i.e., the census block or site-specific). For users with better local elevation and flood and building information, the ability to import such data into the model is critical.

*Prioritization of Census Blocks/Watersheds for Future Mitigation Outreach and Projects*: Floodplain managers need a tool that allows them to view the entire watershed of concern and identify areas that should be targeted for public outreach campaigns. These campaigns usually include promoting residential mitigation efforts, purchasing flood insurance, and, in some cases, participating in voluntary buy-out programs.

*Flood Protection Measures*: Many jurisdictions nationwide have either levees or upstream storage basins as flood protection and other jurisdictions regularly consider the installation of these protective measures. These jurisdictions need a simple tool to allow them to examine the benefits of such measures and to identify or highlight potential impacts throughout the watershed. Those communities already protected by such measures may wish to examine potential impacts resulting from the failure of the protection. Some may even wish to examine the impact of removing the protective measures.

Private corporations could perform an analysis to determine if the installation of flood protection measures at various critical sites is cost effective or whether relocation may be a better flood protection approach.

**III.1.8 User Environment**

The user environment will vary depending on the size, structure, and function of the jurisdictions for which they work. In some agencies, floodplain management, land use planning, and the enforcement of building codes and regulations are highly centralized and managed by only a few people. In other agencies, these functions are decentralized and extensive coordination is required to effectively manage them. Still other agencies rely on consulting companies to provide these services.

Local jurisdictions have primary responsibility for managing development within the floodplain, adopting and enforcing building codes and complying with state and federal regulations; however, it is difficult to specifically identify the number of people involved in completing a task. The numbers vary with specific projects and the political sensitivity of land use planning and management decisions. Regulations require each jurisdiction to assign the responsibility of floodplain management to a single person.

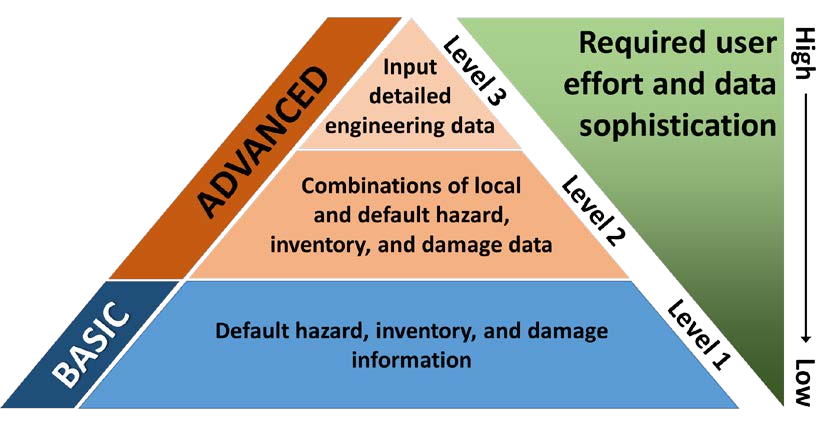
Whether these individuals will actually perform a Hazus analysis will depend on the jurisdiction’s organizational structure. In Hazus training programs, FEMA has taken the approach of recommending that loss estimation be handled by a team of people within an agency. This type of coordination is designed to bring all resources necessary to modify databases, integrate hazard data, review development patterns and trends, general planning, etc.

Hazus supports a wide variety of project cycles. Users interested in loss estimation during an event will obviously prefer fast, accurate results, but the more typical application will allow the user to obtain locally specific data, input that data into the model, and produce results. A typical analysis—including the compiling of data (e.g., digital elevation models, flood study cross sections and elevations, and surveys of structures)—may take several months.

Users currently have several models that help them analyze floodplains. Examples include HEC-2, HEC-RAS, and SLOSH. Most of these software packages identify stream or coastal inundation, flow direction, rates, and other physical flooding characteristics. None help the user develop direct estimates of losses to the exposed building inventory, vehicles, or agriculture products.

**III.2 Levels of Analysis**

To provide flexibility, losses are estimated based on the accuracy of input data. Basic analysis can be based on default data and parameter data provided within Hazus. Advanced analysis can be estimated using more accurate data that is specific to the region, hazard, population, etc. thus improving inventories and/or parameters with user-supplied data. The advanced level also incorporates data from third-party studies. The appropriate level of analysis must be determined to meet the needs and resources of the user.



*Figure 3.8. Levels of Hazus analysis.*

**III.2.1 Analysis Based on Default Information**

A Basic Level 1 analysis is the simplest type of analysis requiring minimum effort by the user. It is based primarily on the national baseline general building stock (GBS), essential facility and demographic databases built into the model. These databases are derived from national-level data sources for building square footage, building value, population characteristics, costs of building repair, and economic data. Default datasets of surface roughness and tree coverage derived from national land-use data are also used for hurricanes. Direct economic and social losses associated with the general building stock are computed, as well as estimates of essential facility functionality, short-term shelter requirement, and debris.

The user is not expected to have extensive technical knowledge. Other than defining the study region, specifying the hazard (probabilistic or scenario), and making decisions concerning the extent and format of the output, an analysis based on default data requires minimal effort from the user. This level of analysis is suitable primarily for preliminary evaluations and comparisons of losses at regional levelss.

**III.2.2 Analysis with User-Supplied Inventory**

Level 2 analysis improves Level 1 results by considering additional data that are readily available or can be easily converted or computed to meet methodology requirements. In Level 2, the user may need to determine parameters from published reports or maps as input to the model. It requires more extensive inventory data and a higher level of effort by the user than a Basic Level 1 Analysis. The purpose of this type of analysis is to provide the user with the best estimates of hazard input data that can be obtained using the standardized methods of analysis included in the methodology. For example, user-supplied flood depth grids should be used over the internal Hazus hydrology and hydraulics model. It is likely that the user will need to employ consultants to assist in the implementation of certain methods. For example, a local geotechnical engineer would likely be required to define soil and ground conditions. Improved results are highly dependent on the quality and quantity of improved inventories. The significance of the improved results also relies on the user’s analysis priorities. The following inventory improvements impact the accuracy of analysis provided by the respective hazard models, as well as applications of the results:

* Use of locally available data or estimates concerning the square footage, count, and replacement values of buildings in different occupancy classes, including the use and integration of site-specific data.
* Use of local expertise to modify the databases concerning percentages of model building types associated with different occupancy classes.
* Preparation of a detailed inventory for all essential facilities and facilities housing hazardous materials.
* Collection of detailed inventory and cost data to improve evaluating exposure of various transportation and utility lifelines.
* Collection of detailed population and transportation data to improve the displaced households, shelter needs, evacuation and casualty results.
* Use of locally available data concerning construction costs or other economic parameters.
* Development of flood, earthquake, wind and tsunami hazard map data. These maps would be used for evaluation of the effects of local conditions upon damage and losses. Higher quality results are produced through the accurate intersection between hazard and inventory.
* Synthesis of data for evaluating the economy of the study region used in assessing indirect economic impacts.

Depending upon the size of the region and the number of these features selected by the user, months may be required to assemble the required input. The effort put into preparing the inventory of the building stock can range from minimal to extensive, depending upon the desire to reduce uncertainty in computed results.

**III.2.3 Analysis with Advanced Data**

A Level 3 analysis requires effort by the user to develop and update information concerning the underlying engineering and loss analysis parameters in Hazus. This type of analysis incorporates results from engineering and economic studies that are typically regionally specific and carried out using methods and software not included within the methodology. At this level, one or more technical subject matter experts are required to acquire data, perform detailed analyses, assess damage/loss, and assist the user in gathering and applying custom damage functions and parameters. There are no standardized Level 3 analysis approaches. It is anticipated that at this level there will be extensive participation by local utilities and owners of special facilities. Users must understand where, within the Hazus software, to change the underlying engineering and loss parameters used for an analysis. The quality and detail of the results will depend on the level of effort.

**III.2.4 Anticipated breakdown of Open Hazus users**

Once OpenHazus is available in the form of web services with a substantial building-level inventory and widely accessible high-resolution depth grids, we anticipate that:

1. 50% of users will primarily want to work with model results, i.e., view reports and maps of results
2. ~5% will require a capability to support the analysis of sensitive or protected data, such as within their own desktop environment
3. 45% will want to replicate, customize and improve upon OpenHazus capabilities with all its cloud advantages

Designing OpenHazus to serve this user breakdown obviates the previously described “Levels of Analysis” originally prescribed for Hazus users. If baseline Hazus model results are provided in a user-friendly online platform, OpenHazus analytical modules will only be run by users seeking to leverage some combination of customized damage functions, inventory data, or hazard data specific to a scenario or community. An actuary may use default inventory and nationwide probabilistic hazard data, but customize the damage functions available in OpenHazus. A local planner might use default OpenHazus damage functions, but upload custom scenario-based hazard data and updated site-specific inventory information to use as model inputs. Since any user running OpenHazus will be using some kind of customized model input, distinguishing between “levels” of customization would overcomplicate OpenHazus implementation.

**III.3 Typical Workflows**

While the overall workflow of Hazus follows the same general logic, there are some differences based on hazard type. The common logic is represented by the following seven steps:

1. Hazard-specific building stock definition
2. Inventory database aggregation
3. Scenario hazard definition and discovery
4. Building damage and loss function parameter display
5. Loss category (module) selection
6. Loss estimation (analysis)
7. Results exporting and display

The following sections highlight the differences between models as a function of hazard type.

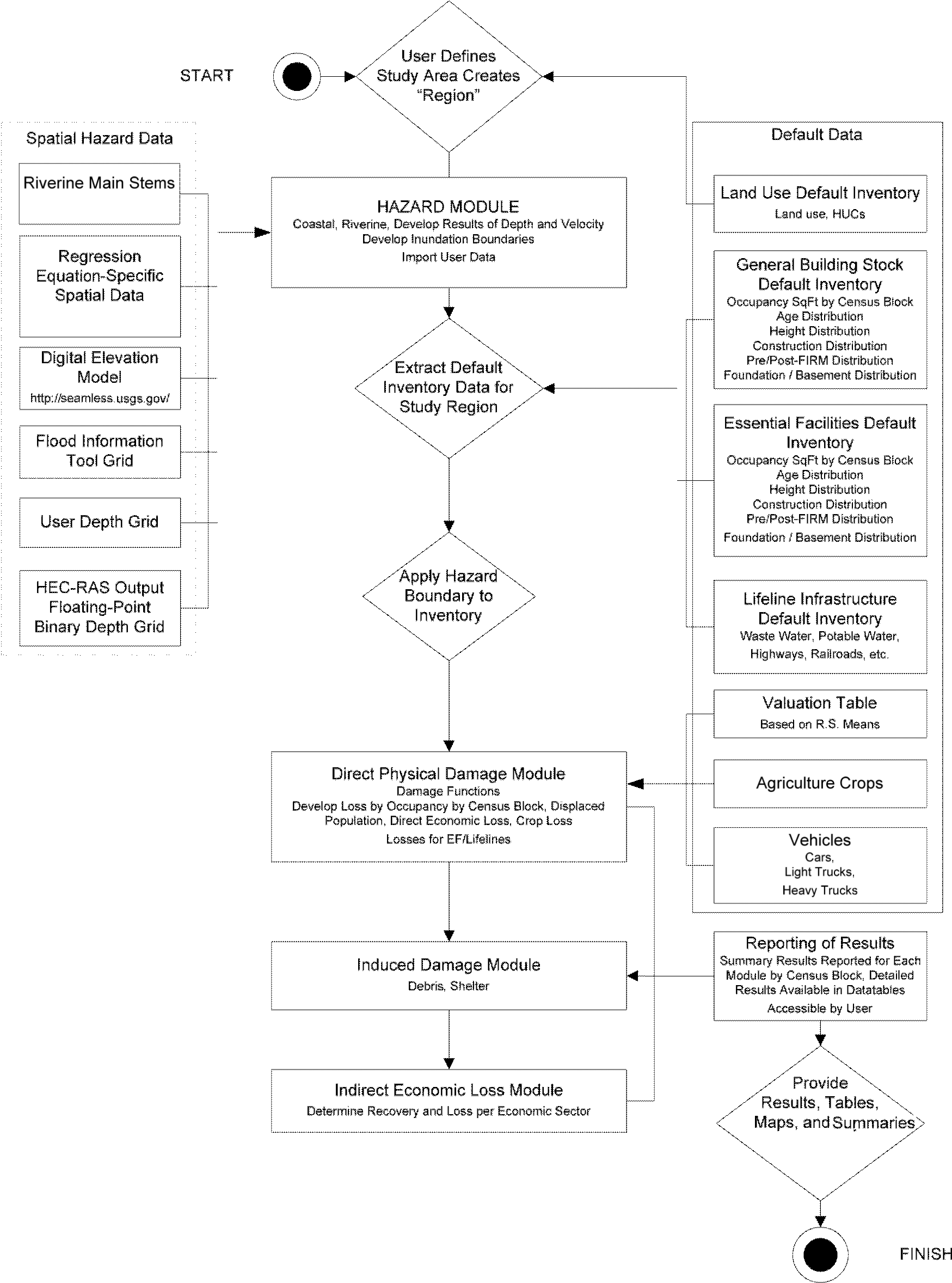
**III.3.1 Flood Model Workflows**

The steps used in the Flood Model are as follows:

* Select the area to be studied. The region of interest is created based on census tract, census block, county, state/territory, community, or watershed. The area generally includes a city, county, or group of municipalities. It is generally desirable to select an area that is under the jurisdiction of an existing regional planning group.
* Specify the hazard. In the Flood Model, the hazard can be specified as riverine, coastal, or a combination of riverine and coastal. In addition, the user can specify an externally created hazard (depth grid) or utilize the Hazus Level 1 H&H capabilities.
* Provide additional information describing the building inventory, or Essential Facilities, if available.
* Using formulas embedded in Hazus, Hazus computes expected building losses, expected contents losses, and expected loss-of-use for different classes of buildings, as well as some social impacts.

Hazus uses the above results to compute estimates of direct economic loss to buildings and infrastructure, as well as displaced households and short-term public shelter needs.

**III.3.1.1 Riverine Flood Level-1 Analysis for General Building Stock**

1. Open a Region
2. Select Flood Hazard Type (riverine vs coastal[[1]](#footnote-1))
3. Chose DEM
4. Develop (synthetic) Stream Network (from DEM)
   1. Define Drainage Area
5. Create a Scenario
   1. Select River Reaches (subset of the above stream network)
   2. If coastal hazard, select shorelines and breaklines
      1. Identify vertical datum
      2. Include optional wave setup
6. Run Hydrologic Analysis
   1. Simulate levees
   2. Determine impact of flood velocity and flow regulation measures
7. Delineate Floodplain (Hydraulic Analysis)
   1. Chose Analysis Type
      1. Single return period (choice of period span)
8. Compute Losses within the Floodplain
   1. Select GBS Damage and Loss Options
      1. Select or modify building depth-damage functions
      2. Restoration functions
   2. Define parameters for
      1. Debris, Shelter, Agriculture, Direct Economic Loss, and Lifelines
   3. Flood Warning (USACE Day curve)
   4. Annualized loss
   5. Combined Wind and Flood (if the user has run the Hurricane model first)

*Figure 3.1.  
Flood Model Schematic consisting of two basic analytical processes:  
① hazard analysis and ② loss estimation analysis*

**III.3.2 Hurricane Model Workflows**

Several steps are typically performed in assessing and mitigating the impacts of a natural hazard such as a hurricane. The methodology encompasses inventory collection, hazard identification, and impact assessment. In a simplified form, the steps are:

* Select the area to be studied. This may be a state, city, a county or a group of municipalities. It is generally desirable to select an area that is under the jurisdiction of an existing regional planning group.
* Specify the hazard. In the Hurricane Model the hazard can be specified as either a single historical or user-defined storm scenario or as a complete probabilistic analysis. When a single event scenario is chosen, the option of developing coastal storm surge and wave estimates is available. These results can be fed into the Hazus Flood Model to produce combined wind and surge loss estimates for the General Building Stock for the surge impacted region.
* Provide additional information describing the building inventory, essential facilities, tree coverage, and surface roughness, if available.
* Using formulas embedded in Hazus, damage probabilities, expected building losses, expected contents losses, and expected loss-of-use are computed for different classes of buildings.
* The above results are used to compute estimates of direct economic loss, essential facility damage and loss, as well as displaced households and short-term public shelter needs.

Using formulas embedded in Hazus, the expected amounts and types of debris are estimated.

**III.3.2.1 Developing data for a complete loss estimation study**

1. Developing a Regional Inventory
   1. Locations of government facilities such as military installations and government offices
   2. Tax assessor’s files
   3. School district or university system facilities
   4. Databases of fire stations or police stations
   5. Databases of historical buildings
   6. Databases of churches and other religious facilities
   7. Postal facilities (ATC-26, 1992)
   8. Hospitals (The AHA Guide of the American Hospital Association; ATC-23A, 1991)
   9. Public and private utility facility databases
   10. Department of transportation bridge inventory
   11. Dun and Bradstreet database of business establishments
   12. Insurance Services Office’s files of large buildings that are used for fire assessment real estate databases
2. Standardizing and Classifying Data
   1. Collecting Inventory Data
      1. General Building Stock
         1. Square footage
         2. Building count
         3. Dollar exposure
      2. Essential facilities
      3. High Potential Loss Facilities
      4. User Defined Facilities
      5. Transportation Systems
      6. Utility Systems
      7. Hazardous Materials
      8. Demographics
3. Building or Modifying Inventory Databases
   1. Importing Features and Files
      1. Importing site-specific data files
      2. Import Database Utility
   2. Adding Records to Site-Specific Databases
      1. Adding features
      2. Adding, editing, deleting records to attribute table
4. Inventory Requirements
5. Relationship between Building Types and Occupancy Classes
   1. Defining Specific Occupancy to General Building Type Mapping Schemes
   2. Defining Specific Building Type Mapping Schemes
   3. Defining Wind Building Characteristics Distributions
   4. Applying Mitigation to the General Building Stock
   5. Defining Wind-Related Building Characteristics for Essential Facilities

**III.3.2.2 Running Hazus Hurricane with user-supplied data**

1. Define the Study Region
2. Define the Inventory Data
3. Define the Hurricane Hazard
   1. Probabilistic
   2. Deterministic
      1. Define storm track
      2. Import storm track
      3. Selecting a historic storm
      4. Import from Census Tract Centroid .dat file (eg. H\*Wind)
      5. Import Hurrevac storm advisory
   3. Review the Currently Defined Hazard
4. Review the Damage, Loss and Debris Functions
   1. Building damage
   2. Building and contents loss
   3. Loss of use
   4. Debris
5. Setting the Analysis Parameters
   1. Defining Tree Coverage Data
   2. Defining Terrain Data
   3. Defining Shelter Parameters
      1. Development of input for displaced households
      2. Fraction of dwelling units likely to be vacated if damaged
      3. Percentage of HH affected by utility outages likely to seek alternative shelter
      4. Development of input for shelter needs
         1. Number of people in the census tract
         2. Number of households in census tract
         3. Income breakdown of households in census tract
         4. Ethnicity of households in census tract
         5. Percentage of homeowners and renters in the census tract
         6. Age breakdown of households in census tract
   4. Defining the Buildings Economic Parameters
      1. Types of direct economic loss
      2. Development of input for building losses
         1. Building replacement values by census tract for all occupancies
         2. Contents values by census tract for all occupancies
         3. Annual gross sales or production in $ per square foot for agricultural, commercial and industrial occupancies
         4. Business inventory as a percentage of gross annual sales for agricultural, commercial and industrial occupancies
         5. Business inventory damage as a function of damage state for agricultural, commercial and industrial occupancies
         6. Building cleanup and repair time in days as a function of wind building type
         7. Rental costs
         8. Disruption costs
         9. Percent of buildings that are owner occupied for each occupancy class
         10. Capital-related income and wage income in $/day per square foot for each occupancy
      3. Building replacement costs
      4. Building contents
      5. Business inventory
      6. Repair and cleanup times
      7. Relocation expenses
      8. Capital-related expenses
   5. Selecting Analysis Options and Running an Analysis
      1. Storm surge options

**III.3.3 Earthquake Model Workflows**

Steps in assessing and mitigating losses due to natural hazards shows the steps that are typically performed in assessing and mitigating the impacts of an earthquake. The methodology incorporates inventory collection, hazard identification, and the natural hazards impact assessment. In a simplified form, the steps include:

* Select the area to be studied. The region of interest is created based on Census Tract, county, or state. The area generally includes a city, county, or group of municipalities. It is generally desirable to select an area that is under the jurisdiction of an existing regional planning group.
* Specify the scenario earthquake. In developing the scenario authoritative data from the USGS ShakeMap program is available through an API.
* Provide additional information describing local soil and geological conditions, if available. Soil characteristics include site classification according to the National Earthquake Hazard Reduction Program (NEHRP) and susceptibility to landslides and liquefaction (ground failure).
* Using formulas embedded in Hazus, probability distributions are computed for damage to different classes of buildings, facilities, and lifeline system components. Loss-of-function is also estimated.
* The damage and functionality information is used to compute estimates of direct economic loss, casualties and shelter needs.

An estimate of the number of ignitions and the extent of fire spread is computed. The amount and type of debris are estimated.

**III.3.3.1 Running Hazus Earthquake with user-supplied data**

1. Define Study Region
2. Define Hazard
   1. Scenario earthquake (deterministic hazard),
      1. USGS ShakeMap scenario or actual earthquake
      2. Historical epicenter event
      3. Source event (selecting source from Hazus faults database)
      4. Arbitrary event
   2. Probabilistic seismic hazard analysis (for eight return periods)
      1. Annualized loss calculation
   3. User-supplied map of ground motion
      1. User supplies digitized peak ground acceleration and velocity, and
      2. Spectral acceleration contour maps
      3. Potential ground displacement (PGD) due to lateral spreading from landslide
      4. Potential ground displacement (PGD) due to settlement from liquefaction
      5. Potential ground displacement (PGD) due to surface fault rupture
   4. Include (soil) site effects using a soil map
   5. Include ground failure using landslide and/or liquefaction susceptibility maps
3. Direct Physical Damage Analysis
   1. Structural vs non-structural
   2. Definition of (five) damage states
      1. Fragility curves (default values)
      2. Fragility curves (user-defined)
   3. Calculating damage state probabilities
   4. Modifying capacity curves
   5. Restoration time
4. Induced Physical Damage
   1. Fire
      1. Fire parameters
      2. Number of simulations
   2. Hazardous materials facility exposure
   3. Debris estimate
5. Direct Social and Economic Loss
   1. Casualties
   2. Displaced households/public shelter needs
   3. Direct economic loss
      1. Development of input for building losses
      2. Replacement costs
      3. Business inventories
      4. Cleanup and repair time
      5. Relocation expenses
      6. Capital-related expenses
6. Indirect Economic Loss
   1. For each of ten sectors
   2. Run loss model with a synthetic economy
7. Incorporating Uncertainty in Reporting Results

**III.3.4 Tsunami Model Workflows**

Several steps are typically performed in assessing and mitigating the impacts of a tsunami. In a simplified form, the steps include:

1. Select the area to be studied. The region of interest is created based on Census block, Census tract, county, NFIP community, or state. The area could include a city, county, or group of municipalities. It is generally desirable to select an area that is under the jurisdiction of an existing regional planning group.
2. Specify the hazard. In the Tsunami Model, the hazard can be specified as a Near Source or Distant Source tsunami. When a Near Source scenario is selected, the option of running a combined earthquake and tsunami damage scenario is available. The results of the earthquake shake damage can be fed into the Hazus Tsunami Model to produce combined earthquake and tsunami loss estimates for the General Building Stock (GBS) and User-Defined Facilities (UDF).
3. Select an analysis level (Runup-only, depth & velocity, depth & momentum flux)
4. Choose and mask DEM if required (not required if user has depth and velocity data)
5. Provide additional information describing the building inventory, essential facilities, and demographics, if available. Hazus tsunami does not require tsunami-specific inventory attributes, rather it leverages the existing earthquake and flood attributes.
6. Using formulas embedded in Hazus, damage probabilities, expected building losses, expected contents losses, and expected loss-of-use are computed for different classes of buildings and for user-defined facilities.
7. The results of the previous step are used to compute estimates of direct economic loss, evacuation times, and casualties.

**III.3.4.1 Running a combined earthquake and tsunami model**

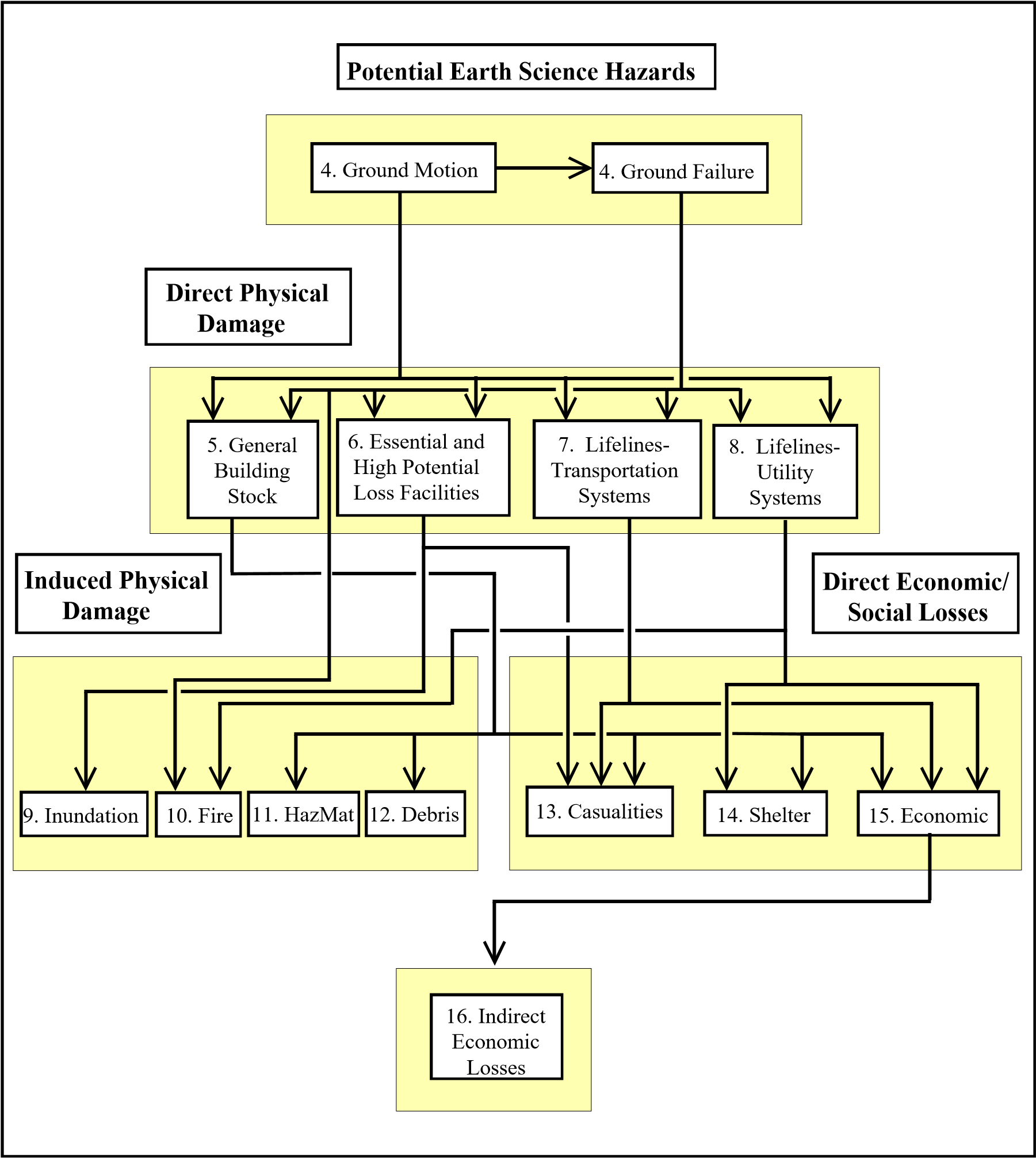
|  |  |
| --- | --- |
| ***Earthquake Model*** | ***Tsunami Model*** |
| * Define/Select Earthquake Scenario  (using same scenario source that creates the Tsunami hazard) * Run Analysis * Display Earthquake-Only Losses | * Select Tsunami Scenario * Define Tsunami Type as Near Source * Define Scenario – Level 1, 2 or 3 * Run Tsunami Analysis * Define Casualty Level 1 or 2 * Display Combined Earthquake and Tsunami Losses |

**III.3.4.2 Pedestrian evacuation workflow example**

1. Create/set a portfolio for the study area
2. Preprocess data
   1. Preprocess DEM and roadway data (or LULC)
   2. Preprocess hazard
   3. Validate safe zone
3. Create surfaces and maps
   1. Calculate least-cost distance path
   2. Create evacuation travel-time surface
   3. Determine max time value for each Census Block
   4. Create time map
4. Process vertical evacuation sites
   1. Process vertical evacuation sites
   2. Merge safe zones
5. Population (Day/Night & Over/Under age 65) processing

**IV Solution Framework**

The vision of OpenHazus loss estimation requires a methodology that is both flexible –accommodating the needs of a variety of different users and applications – and standardized such that model results are somewhat uniform, understandable, and trusted. The framework of the methodology includes each of the components shown in Figure 4.1. As indicated by arrows in the figure, modules are interdependent with output of some modules acting as input to others. In general, each of the components will be required for loss estimation.



*Figure 4.1. Flowchart of the Hazus loss estimation methodology.*

However, the degree of sophistication and associated cost will vary greatly by user and application. It is therefore necessary and appropriate that components have multiple levels (multiple modules) of detail or precision when required to accommodate user needs. The approach is based on a hazard-load-resistance-damage-loss methodology developed from an individual risk framework (Li & Ellingwood 2009[[2]](#footnote-2)). The basic model components (hazard model, load model, resistance models, etc.) are developed separately. Each model component is, wherever possible, separately validated using full-scale data, model scale data, or experimental data. A first principles-based hazard-load-resistance-loss model is used, providing the capability to model the effects of building code changes and mitigation strategies on reduction in damage and loss. Furthermore, since economic damage (loss) is modeled separately from physical damage to a building, estimates of both building damage and loss are separately modeled and predicted.

**IV.1 Modularization**

OpenHazus utilizes a modular approach with different modules addressing different analytical processes users may require to complete their risk assessment projects. This approach avoids the need to designate a fixed set of OpenHazus users. The needs of most, if not all, current and potential users can be accommodated by the flexibility of fully encapsulated analytical “tools” available within the larger framework of an online OpenHazus “toolbox.”. A modular analytical framework also reduces the training burden on users, who are no longer required to understand all components of a large, rigid workflow, but can instead run only those modules that are familiar to their particular industry or applicable to their particular data inputs. This approach also permits users to select methods (modules) that produce varying degrees of precision appropriate for their region for either deterministic or probabilistic risk assessments.

Framing the hazard loss estimation methodology as a collection of modules permits adding new modules (or improving the methods or data of existing modules) without reworking the entire methodology. Fundamental OpenHazus modules accept user-designated maps of depth grids, wind fields, earthquake ground shaking or tsunami depth and velocity to apply damage functions and assign losses for user-designated inventories. However, improvements may be made to adapt modules to local or regional needs, to incorporate new hazards and data inputs, or to experiment with new methods for calculating hazard impacts. All of these OpenHazus module updates can be implemented as resources become available, rather than within a rigid update cycle. OpenHazus methodologies can evolve logically as research progresses and the state-of-the-art advances. Module-by-module development also provides a feasible funding approach for a multi-year transition to OpenHazus.

**IV.2 Independence from Desktop GIS**

While Hazus has so far relied on GIS as the framework from within which individual modules are called, OpenHazus is designed to be GIS-independent. This approach is driven by the fact that a tiny fraction of current Hazus analytical processes are uniquely geospatial. This small fraction includes traditional desktop GIS functions for clipping, mapping, projecting, and developing flood-related rasters. Geoprocessing steps dedicated to developing flood hazard data will be eliminated by the OpenHazus emphasis on external authoritative hazard datasets as model inputs. The remaining relatively simple geospatial analysis functions can be easily replaced by open source geospatial libraries that meet the same analytical needs independent from a traditional desktop GIS architecture. These open source geospatial analysis tools can be run in a cloud environment or on a local machine through a lightweight, downloadable OpenHazus utility.

The relatively insignificant role of geoprocessing steps in Hazus model design indicates that the primary function of desktop GIS software in Hazus is the spatial visualization of data inputs (inventory, hazard) and model results (losses). As spatial and non-spatial data analytics become increasingly integrated, mapping functionality is more logically provided to users under a larger umbrella of data reporting tools. Hazus spatial visualization functions are better accomplished through open source mapping libraries that can be seamlessly bundled with non-spatial data visualization libraries in one intuitive “toolkit” that provides OpenHazus users with diverse options for interpreting and representing their model results. Traditional desktop GIS technology may still be used by stakeholders for preparing advanced inventory or hazard data, as well as subsequent analysis and interpretation beyond the scope of Hazus.

Decoupling OpenHazus from traditional desktop GIS technology carries the potential to dramatically widen its user base by removing the extreme specialization often associated with spatial data analysis. As data analysis skillsets become more common throughout disaster-related disciplines, the risk modeling community has grown to include those who are interested in estimating hazard impacts but are unfamiliar with traditional GIS technology. The unnecessary reliance of Hazus on desktop GIS software alienates these potential users. A more agnostic risk modeling platform – one that seamlessly incorporates both spatial and non-spatial components – will address the needs of a much broader user base.

It is important to note that the de-emphasizing of GIS-specific functionality in OpenHazus should not be mistaken for a de-emphasizing of technical skillsets among users in general. Indeed, the traditional GIS architecture of Hazus often obscures the technical complexity of hazard risk modeling such that users with GIS specializations are expected to undertake risk assessment projects that require professional or academic expertise in engineering, earthquake physics, structural dynamics, and advanced statistics. A GIS-independent OpenHazus architecture will still require technical expertise in order to accurately produce and interpret certain model results. However, the technical requirements of each analytical module will be more exposed to users in an open source, modular framework, so that users can more effectively leverage those modules that fall within their professional purview. Increased transparency of technical model requirements coupled with the OpenHazus baseline results platform will more effectively address user needs while reducing the prevalence of erroneous model results common in GIS-based Hazus.

The modular approach includes standard methods for:

* + - * 1. Inventory data collection and updates by attribute type
        2. Using database maps of soil type, ground motion, ground failure, etc.
        3. Using database maps of terrain elevations
        4. Classifying occupancy of buildings and facilities
        5. Classifying building structure type
        6. Describing damage states
        7. Developing building damage functions
        8. Grouping, ranking and analyzing lifelines
        9. Using technical terminology
        10. Visualizing and interpreting model results

**IV.2 Advantages of the Open Character**

The current Hazus hazard loss estimation methodology is an improvement over existing regional loss estimation methodologies because it is more comprehensive and inter-disciplinary than local single-hazard studies. Examples of these impacts are service outages for lifelines, estimates of fire ignitions and fire spread, potential for a serious hazardous materials release incident, and indirect economic effects. One important aspect in which OpenHazus goes beyond the capabilities of Hazus is its ability to run scenario ensembles and to employ modern visualization techniques that help stakeholders sort through the vast amounts of data generated. The ability to ask “what if” questions – and get them answered – will be vastly improved, especially for organizations that do not have the computational facilities locally available that the web services-based solution of OpenHazus provides.

Another advantage of OpenHazus is the ability of stakeholders to add to the methodology, which has been a costly and time-consuming project in the past. For instance, the current Hazus methodology calculates potential exposure to flood (e.g., dam break) or fire (following earthquake) in terms of the fraction of a geographical area that may be flooded or burned but does not have methods for rigorous calculation of damage or loss due to flooding or fire as secondary earthquake hazards. Consequently, these two potential contributors to the total loss are now obvious foci of community-developed modules for estimations of economic loss, casualties or loss of shelter. Another example is the incorporation of the USGS Pedestrian Evacuation Analyst software into Hazus tsunami. While this was of significant benefit when initially developing Hazus tsunami, incorporating future USGS enhancements will be costly and inefficient without a modular framework.

**IV.2.1 Training and Outreach**

**IV.3 Inventories as an Indispensable Aspect of Open**

Developing the high-quality data required for an accurate and applicable risk assessment is often cost-prohibitive. The collection of inventories is without question the costliest part of performing a risk study. Furthermore, many municipalities have limited budgets for performing a multi-hazard loss estimation study, but often have good inventory data to integrate. It is thus of utmost importance that high-quality inventory data are readily accessible and easy to update with users data to accommodate different users with different levels of resources.

While most users will develop a local inventory that best reflects the characteristics of their region, such as building types and demographics, OpenHazus is designed to provide reasonable estimates of losses based on a minimum of local input. We have a come a long way since the first releases of USGS or U.S. Census data. Open data are more than a buzzword and has become the basis for many business processes, including those of governments. The quality and richness of user-generated geospatial data such as Open Street Map has been studied and proven. (Sehra et al. 2014, Basiri et al. 2016, Barrington-Leigh & Millard-Ball 2017)[[3]](#footnote-3). Private entities like Google and Microsoft now provide web services with well documented APIs and monthly as well as needs-driven updates down to the building level (Microsoft 2018[[4]](#footnote-4)) based on imagery. In addition, the advent and prevalence of lidar data have resulted in accurate building footprint data over most urban areas.

Some analyses require data that will for security reasons never be made public. For example, a stakeholder may wish to perform detailed assessments of damage to and service outages for lifelines, which requires cooperation and input from utilities and transportation agencies. Lifeline systems require an understanding of the interactions between components and the potential for alternative pathways when certain components fail. Thus, without cooperation of utilities, the user is limited in the quality of analysis that can be performed. OpenHazus will need to provide options to either upload such data to a private/secure space alongside publicly accessible inventory data or (the 5% of Open Hazus users dealing with PII data referred to in section III.2.4 of this whitepaper) to link to such data locally.

**IV.3.1 Requirements for Demographic Data**

The primary applications of demographic data are for displaced households, public shelter needs and casualty modelling. Certain demographic attributes are used more than others, for example, age, ethnicity, homeownership, and income are used to estimate the numbers of displaced individuals likely to seek public shelter. Employment and household’s data are used to estimate the distribution of people during the day and night, as well as commute time for casualty modelling. Income as a ratio of the national median is used to estimate the construction quality of single-family residences for building replacement cost purposes. The current Hazus process of updating demographic data requires waiting several years after the decennial census and implementing a manual update process. However, a new design can leverage more dynamic data available through the annual American Community Survey and Longitudinal Employer-Household Dynamic datasets.

In addition to traditional tabular access, the Census Bureau now offers a robust API to (as of time of writing[[5]](#footnote-5)) provides 327 datasets, often in the Gigabyte range: <https://api.census.gov/data.html>. An alternative for international users, albeit without an API yet, is IPUMS, the world’s largest accessible database of census microdata. IPUMS includes almost a billion records from U.S. censuses from 1790 to the present and over a billion records from the international censuses of over 100 countries. What distinguished IPUMS is its harmonization of survey data with over 30,000 integrated variables and 150 million records, including the Current Population Survey, the American Community Survey, the National Health Interview Survey, the Demographic and Health Surveys, and an expanding collection of labor force, health, and education surveys. In total, IPUMS currently disseminates integrated microdata describing 1.4 billion individuals drawn from over 750 censuses and surveys.

The advantage of the US Census API is that it does not require CDS to store the data on its own servers and that the data is always up-to-date. Non-US users will have to create snapshots of the IPUMS data but then gain the advantage that comparisons across changing census geometries are a lot easier than for the original US Census data.

While there are commercial demographics data providers, they tend to be expensive and provide no further detail than the original Census data. Similarly, non-for-profit organizations and large Internet web service providers such as Amazon, Microsoft, or Google have no indigenous data collection efforts of their own but provide only commonly accessed tables that minimize data transfer needs.

One other useful source of demographic data is the disaggregated (rasterized) datasets developed at ORNL and CIESIN. ORNL’s [LandScan](https://landscan.ornl.gov/)™ data comes at 1 km resolution, which is the same as CIESIN’s gridded Population of the World, however, CIESIN’s [Hazards Mapper](http://sedac.ciesin.columbia.edu/mapping/hazards/) adds an OGC web processing service that calculates affected populations as a function of earthquakes, fires, floods, and nuclear plant failures. Both datasets have been used for further disaggregation to 100 m resolution in the Worldpop archive (Lloyd *et al*. 2017)

**IV.3.2 Requirements for Building and Critical Infrastructure Data**

HSIP Open Essential Facility data provide a new dynamic update opportunity for OpenHazus. Currently, Hazus contains baseline essential facility inventories for the U.S. provided in downloadable State databases in SQL format. These baseline inventories provide the user an out of the box capability to run Hazus analyses anywhere in the U.S. However, updating these static data and disseminating to users could be a more efficient and a less manual process. As a result, the lag times associated with updates mean some of the datasets are old and do not benefit from the latest [HIFLD Open](https://hifld-geoplatform.opendata.arcgis.com/datasets) and other releases.

OpenHazus should leverage HIFLD Open, developing scripts, tools and support data to integrate and assign the attributes required for Hazus loss modelling dynamically. We should explore and develop innovative solutions for dynamic data updates that can operate within the future OpenHazus platform deployed in a web environment. For example, if a user is working on a study region online, are there innovative ways to connect to and include HIFLD data dynamically or will we need to download and process data. If not updated dynamically, would like to strive for ways to automate the preparation of the data so updates could be very frequent.

Open HAZUS will build and expand on this as there is clearly a need for building-level data. As of today, such data is limited to ad hoc requests to ORNL for building outlines based on recent imagery. Locally, some well-funded agencies (like New York City OEM) started to combine building permit, tax assessment, and employment data to describe building contents and populations for more than two dozen building occupancy classes. Similarly, some agencies have day-by-day accounts of how many people are, say in a hospital or in a particular school.

Creating such datasets on a nation-wide scale will require significant coordination efforts. An example for that is Google’s [General Transit Feed Specification](https://developers.google.com/transit/gtfs/) (GTFS), which allows local agencies to provide standardized feeds of transit updates, which in turn can be consumed by any interested party to create optimized routing information. While this may be of interest for evacuation purposes, the bigger lesson here is that with proper standards, there is an incentive for local agencies to generate feeds that can be consumed by Open HAZUS (and others) for highly detailed building and infrastructure data.

GTFS is a nice example for how to setup suitable web services. But for the detailed description of building and infrastructure data, a more suitable set of standards is the OGC’s [CityGML](https://www.opengeospatial.org/standards/citygml) as an application schema of the more generic [Geography Markup Language](https://www.opengeospatial.org/standards/gml) or GML, version 3, and the [3D City Database](https://www.3dcitydb.org/3dcitydb/3dcitydbhomepage/), for which there are Oracle and PostGIS tools available that make it eminently suitable for Open HAZUS. The 3DCityDB web feature service allows for feature-level analysis in both flood and earthquake models.

**IV.3.3 Requirements for Depth Grids**

The single biggest obstacle to a successful implementation of Open HAZUS is the lack of a sufficiently detailed depth grids for the United States. USGS began in 2015 with the production of a 1-meter DEM (Arundel et al. 2015) based on NAVD 88 and NAD 83. The vertical accuracy aims to be eventually 6 cm - 8 cm but this requires LIDAR coverage, which so far exists for no more than about ½ of the country (excluding Alaska)[[6]](#footnote-6). By the time, Open Hazus will be fully functional, the 3DEP program is likely to be complete.

This will then provide the basis for the development of a nationwide, purely algorithmically-derived level-1 depth grid. Where available, high water mark data should be employed to estimate flood surfaces. In all other instances, MSC should collaborate with the CyberGIS Center at the University of Illinois and USACE to run the latter’s HEC-RAS models to calculate the flood surfaces necessary that will then be subtracted from the 1-meter DEMs to derive the depth grids.

**IV.3.4 Requirements for Results Data**

Hazus has a wide range of users with diverging needs and experience. The latter applies to experiences in the hazard management realm, as well as in insurance mapping or information processing. Desktop users are at liberty to organize their results as it fits their institutional needs but Open Hazus would lose a lot of its attraction if the results were not sharable. Results can be interpreted as the result of model runs as well as the successful development of a scenario, i.e., in addition to traditional data repositories, the results database should also contain a section of process models[[7]](#footnote-7). An example for such a component of the results database is the [Spatial Decision Support Knowledge Portal](http://sdsportal.sdsconsortium.org/). The Open Hazus development team would have to develop a model description template based on the [BPMN](http://www.bpmn.org/) or the UML profile [SysML](https://sysml.org/).

Careful consideration will need to be given to the organization of the online results database. Model results need to be discoverable by location, hazard type, and filterable by model attributes, including the results provider. Many municipal, state and federal agencies provide their open data on the [Socrata platform](https://socrata.com/publica-open-data/), which is generic enough to be replicated for Open Hazus as well. New York City’s open data portal may serve both as an example and a warning how not to implement Socrata. In spite of having dedicated support staff, the data portal is poorly curated, which makes it hard to distinguish useful from useless datasets. Any discoverable result will therefore have to be checked by FEMA staff for compliance with yet to be determined quality standards.

**V Architecture**

Hazus was designed to work on top of ESRI’s ArcGIS Desktop platform. The workflow was hence spatially driven. Hazus functionality was accessed through three additional menus within the ArcMap user interface. With the decision for OpenHazus to be GIS-independent, a number of constraints to the user interface and the necessary sequencing of processing steps will become obsolete. For instance, the definition of a study area is easily accomplished through a web service that displays a zoomable map of the world, pre-set to the extent of the United States, or through a text-based jurisdictional selection. With much of the inventory data (including an expanded set of higher resolution GBS and demographic data) readily accessible on OpenHazus servers (see section IV.3), the software architecture has become much more open and can be interpreted as a library of methods, similar to the original philosophy of the Unix operating system or the open source GIS GRASS.

The importance of having a Cloud-based repository of actionable data cannot be over-emphasized. Open (and web-based) Hazus is only feasible if much of the burden of data generation, especially the computationally-intensive generation of depth grids, is off-loaded to the server side. The remaining steps of hazard definition, damage, loss, and impact estimation, as well as the generation of report elements are easily handled in form of web services. This in turn facilitates OpenHazus as a light-weight or thin client.

In terms of traditional ArcGIS users, OpenHazus now becomes a toolbox with many toolsets (modules) and potentially hundreds of tools that are extendable by the user community. Like the (geo-) processing models in ArcGIS and QGIS, OpenHazus users will be able to design their own workflows and customize OpenHazus to their needs. The Risk MAP CDS team will provide a reference implementation that stakeholders may use out-of-the-box, but the assumption is that interested parties will create their own tools, models, and workflows.

The following pages will describe typical workflows and their associated tools or methods in the proposed OpenHazus architecture. They are organized by hazard and type of analysis, followed by a mapping of tools to types of users (viewers, editors, analysts).

**V.1 Hazard**

**V.1.1 Flood Modules**

**V.1.1.1 Inventory Creation**

The majority of Open Hazus users are expected to work with inventory data that is provided on Open Hazus servers. However, it could not be called open if there was not the facility to create specialized inventories that are tailored to fit the user’s needs. In addition to providing tools for building inventory data similar to the existing CDMS, Open Hazus requires a set of standards for how the respective inventories should be organized and what they should contain.

**V.1.1.1.1 Building and facilities**

Current Hazus inventory descriptions are based on a set of 128 building types developed in the 1990s. This work has been significantly expanded by colleagues (Brzev *et al*. 2013) in the Global Earthquake Model (GEM) project, who developed a taxonomy of building types for worldwide use that has found wide-spread adoption. In this taxonomy, each building is characterized by thirteen variables with mappings from each resulting building type to a range of structural taxonomies world-wide. The GEM building taxonomy has been implemented in building information modelling (BIM) systems and has become a standard in the insurance industry. Its open character suggests easy adoption for Open Hazus as well.

**V.1.1.1.2 Essential and user-defined facilities**

As in its current incarnation as Hazus-MH, Open Hazus should provide the standard HIFLD database of essential facilities. One upside to an online version is that it could be updated as soon as new HIFLD data is made available. Similarly, gold versions of that database could be made available once accessibility is managed on a secure login basis. More important, however, will be the Cloud-based storage of user-defined facility data so that it becomes available to authorized users from a range of mobile devices.

**V.1.1.1.3 Transportation lifelines**

For transportation lifelines, it is necessary to distinguish between an established standard for transit systems and a plethora (/zoo) of standards for road-traffic involving private cars and trucks.

V.1.1.1.3.1 Transit feeds

Most public transit authorities in the US and many others world-wide[[8]](#footnote-8), have adopted the [General Transit Feed Specification (GTFS)](https://developers.google.com/transit/gtfs/) that provides potentially near real time updates on the status of public transit facilities. A number of APIs are made available on a crowd-sourced platform called [transitland](https://transit.land/documentation/).

V.1.1.1.3.2 Emerging standards for intelligent transportation systems

The US Department of Transportation has developed standards for [emergency transportation operations](https://www.standards.its.dot.gov/LearnAboutStandards/ResearchInitiatives#emergency), [integrated corridor management systems](https://www.standards.its.dot.gov/LearnAboutStandards/ResearchInitiatives#integrated), [transportation weather observation and forecasting](https://www.standards.its.dot.gov/LearnAboutStandards/ResearchInitiatives#clarus), and [vehicle infrastructure integration](https://www.standards.its.dot.gov/LearnAboutStandards/ResearchInitiatives#vehicle). It is beyond the scope of this report to make recommendations with respect to which of these should be incorporated into transportation lifelines inventories but future developers of Open Hazus should keep an eye on these initiatives as they have the potential to revolutionize the implementation of individual agent-based modelling of hazard scenarios as well as actual emergency response.

**V.1.1.1.4 Lifeline utility systems**

Past experiences with efforts to collaborate with electricity, gas, phone/internet, pipeline, water/sewer companies on hazard mitigation suggest that this is an area, where the openness of Open Hazus does not add any identifiable benefits. Private as well as public utilities in the United States have been very reluctant to share data even in secure settings, and this situation is unlikely to change until the federal government adopts international best practices (Asia and Europe) of successful collaborations. A case in point is the fact that many of the relevant files in HIFLD have not been updated for ten years.

**V.1.1.1.5 Agricultural products**

Hazus-MH currently uses the US Department of Agriculture’s National Resources Inventory (NRI) and National Agriculture Statistical Service (NASS) datasets. These are updated every three to four years and are usually at county-level resolution (although there are some gaps in the data). At a minimum, these updated tables should be available as part of Open Hazus as well, however, an advisory committee should look into higher-resolution data, both spatially and temporally, that the USDA is using for farm-based subsidies.

**V.1.1.1.6 Vehicles**

The current vehicle damage estimation procedures are excellent and can be applied in their current form to Open Hazus.

**V.1.1.1.7 Hazardous material facilities**

Current Hazus-MH hazmat classifications are based on the 2006 National Fire Protection Association’s Uniform Fire Code. This is (a) outdated, as the NFPA updates its code tables on an annual basis and (b) is limiting because it deals with a rather limited number of hazardous materials. Some municipalities have adopted the wider ranging United Nations’ (and US DoT) recommendations on the transport of dangerous goods (UN 2017[[9]](#footnote-9)), which covers radioactive, flammable, explosive, corrosive, oxidizing, asphyxiating, biohazardous, toxic, pathogenic, or allergenic hazards. There is, however, so far no standardized way to assemble inventories based on the DoT classification. Any modernization effort of Hazus, Open or MH) should include a committee of volunteers to develop such a standard.

**V.1.1.1.8 Direct economic and social losses**

The direct economic loss methodology is an amalgamation of the individual above listed losses (1.1.1.1 to 1.1.1.7) and as such dependent on the recommendations made above. As will be discussed in detail in chapter IX, for Open Hazus modern dashboard tools based on open source libraries such as XY would replace the Crystal report components of the existing Hazus MH.

The methodology for the calculation of casualties has been underdeveloped[[10]](#footnote-10) and there are no immediate plans to change that. Loss of housing habitability and need of short-term shelter are derived from the building inventory and the existing algorithms can be directly applied to any new shape and form that inventory will take as per section 1.1.1.1.

**V.1.1.1.9 Indirect economic losses**

The calculations of indirect economic losses for each hazard are well described in the technical manuals of Hazus-MH but have found little adoption even by well-funded stakeholders. The input-output models (Figure 5.1), even in their simplified form, require data that is usually not available to the emergency management community and a level of economic expertise that is typically reserved to staff with at least a masters degree in economics – not a typical skill in emergency management.

*Figure 5.1. Indirect loss module schematic (from the Hazus MH-Quake technical manual).*

On the data side, the Dun & Bradstreet data has proven to be unreliable – yet, there is no single viable alternative. The authors of this report anticipate that companies like Microsoft or Google (or a yet unknown data science startup) will develop aggregation services that mine publically accessible data to develop databases to fill this gap in the marketplace but it is too early to forecast the price point or the level of detail that such a web service might provide.

On the methods side, the existing indirect loss module assumes input-output tables in [IMPLAN format](http://www.implan.com/software/). This, in turn, requires collaboration with institutions whose mission is long term panning of socio-economic policies – a possible but rare juncture. The methodology is well-tested and the documentation within Hazus-MH is good. In the immediate future, CMS would be well served with dedicated courses or tutorials for this Hazus-MH module; in the long run, Open Hazus provides the opportunity to use open source products like [blueNOTE](https://aae.wisc.edu/BlueNOTE) or the more generic [AMPL](https://github.com/ampl/mp).

**V.1.1.2 Flood Hazard Characterization**

**V.1.1.2.1 Riverine flood**

**V.1.1.2.1 Coastal flood**

**V.1.1.2.1 Combined flood**

**V.1.1.2.1 Other flood types**

**V.1.2 Wind Modules**

**V.1.2.1 Inventory Creation**

**V.1.2.1.1 Building and facilities**

**V.1.3 Earthquake Modules**

**V.1.3.1 Inventory Creation**

**V.1.3.1.1 Building and facilities**

**V.1.4 Tsunami Modules**

**V.1.4.1 Inventory Creation**

**V.1.4.1.1 Building and facilities**

**V.2 Analysis**

**V.3 Mapping of Modules to User Types**

**VI. Product Review**

The purpose of this whitepaper is to define the functionality, external interfaces, performance, attributes, and design constraints of Open Hazus. It is based on a Product Requirements Assessment (PRA) focused on the major capabilities and features needed by its stakeholders. This document will not define additional capabilities but focuses on the provision of the capabilities as defined in the PRA. Because Open Hazus will be distributed for free by the federal government, other topics, such as pricing, competition analysis, marketing issues, are not relevant and are not addressed.

**VI.1 Description of Open Source Desktop GIS**

There are dozens of open source geospatial desktop applications and it is beyond the scope of this whitepaper to describe them all. The following six sub sections are intended to describe well-established, widely supported and implemented GIS packages that fit the scope of Open Hazus. Specialty applications, developed for niche application areas (the [Whitebox geospatial analysis tool](http://www.uoguelph.ca/~hydrogeo/Whitebox/) from the University of Guelph might serve as an example here) as well as applications whose core area is image processing are specifically excluded from the discussion here.

**VI.1.1 Geographic Resources Analysis Support System (GRASS)**

The [Geographic Resources Analysis Support System](https://grass.osgeo.org/), commonly referred to as GRASS GIS, is used for data management, image processing, graphics production, spatial modelling, and visualization of many types of data. It is Open Source released under GNU General Public License (GPL) >= V2. GRASS GIS is an official project of the Open Source Geospatial Foundation.

Originally developed by the U.S. Army Construction Engineering Research Laboratories (USA-CERL, 1982-1995), a branch of the US Army Corp of Engineers, as a tool for land management and environmental planning by the military, GRASS GIS has evolved into a powerful utility with a wide range of applications in many different areas of applications and scientific research. GRASS is currently used in academic and commercial settings around the world, as well as many governmental agencies including NASA, NOAA, USDA, DLR, CSIRO, the National Park Service, the U.S. Census Bureau, USGS, and many environmental consulting companies. The GRASS Development Team has grown into a multi-national team consisting of developers at numerous locations.

GRASS GIS contains over 350 modules to render maps and images on monitor and paper; manipulate raster, and vector data including vector networks; process multispectral image data; and create, manage, and store spatial data. GRASS GIS offers both a graphical user interface as well as command line syntax for ease of operations.

**VI.1.1.1 GRASS GIS Capabilities**

Raster analysis: Automatic raster line and area to vector conversion, Buffering of line structures, Cell and profile data query, color table modifications, Conversion to vector and point data format, Correlation / covariance analysis, Expert system analysis , Map algebra (map calculator), Interpolation for missing values, Neighborhood matrix analysis, Raster overlay with or without weight, Reclassification of cell labels, Resampling (resolution), Rescaling of cell values, Statistical cell analysis, Surface generation from vector lines

3D-Raster (voxel) analysis: 3D data import and export, 3D masks, 3D map algebra, 3D interpolation (IDW, Regularized Splines with Tension), 3D Visualization (isosurfaces), Interface to Paraview and POVray visualization tools

Vector analysis: Contour generation from raster surfaces (IDW, Splines algorithm), Conversion to raster and point data format, Digitizing (scanned raster image) with mouse, Reclassification of vector labels, Superpositioning of vector layers

Point data analysis: Delaunay triangulation, Surface interpolation from spot heights, Thiessen polygons, Topographic analysis (curvature, slope, aspect), LiDAR

Image processing: Support for aerial and UAV images, satellite data (optical, radar, thermal), Canonical component analysis (CCA), Color composite generation, Edge detection, Frequency filtering (Fourier, convolution matrices), Fourier and inverse Fourier transformation, Histogram stretching, IHS transformation to RGB, Image rectification (affine and polynomial transformations on raster and vector targets), Ortho photo rectification, Principal component analysis (PCA), Radiometric corrections (Fourier), Resampling, Resolution enhancement (with RGB/IHS), RGB to IHS transformation, Texture oriented classification (sequential maximum a posteriori classification), Shape detection, Supervised classification (training areas, maximum likelihood classification), Unsupervised classification (minimum distance clustering, maximum likelihood classification)

DTM-Analysis: Contour generation, Cost / path analysis, Slope / aspect analysis, Surface generation from spot heights or contours

Geocoding: Geocoding of raster and vector maps including (LiDAR) point clouds

Visualization: 3D surfaces with 3D query (NVIZ), Color assignments, Histogram presentation, Map overlay, Point data maps, Raster maps, Vector maps, Zoom / unzoom -function

Map creation: Image maps, Postscript maps, HTML maps

SQL-support: Database interfaces (DBF, SQLite, PostgreSQL, mySQL, ODBC)

Geostatistics: Interface to "R" (see section VII.8), Matlab, and other software packages

Temporal framework: support for time series analysis to manage, process and analyze (big) spatio-temporal environmental data. It supports querying, map calculation, aggregation, statistics and gap filling for raster, vector and raster3D data. A temporal topology builder is available to build spatio-temporal topology connections between map objects for 1D, 3D and 4D extents.

Furthermore: Erosion modelling, Landscape structure analysis, Solution transport, Watershed analysis.

**VI.1.2 Quantum GIS**

[Quantum GIS](https://qgis.org/en/site/), better known by its acronym QGIS, is the leading open source ***desktop*** GIS. It allows users to create, edit, visualize, analyze and publish geospatial information on Windows, Mac OS, Linux, BSD and Android (via the QField app). There is also an OGC Web Server application, a web browser client and developer libraries. QGIS is licensed under the GNU General Public License. The QGIS project is under very active development by an enthusiastic and engaged developer community with good mechanisms for help via stack exchange, mailing lists and through a global network of commercial support providers.

QGIS has grown from a simple ArcView-like shapefile manipulation program to a full-fledged GIS that mirrors to a large degree the concepts and functionalities of ArcGIS. With this came a now confusing plethora of UX items that lacks the streamlining that ESRI has gone through in its development from ArcGIS Desktop to ArcGIS Pro. However, if one were to base a desktop implementation of Open Hazus on QGIS, then it would be possible for developers to customize the user interface to the specific needs of a hazards-oriented spatial decision support system.

QGIS connects the spatial versions of Postgres, SQLite, MS SQL, and Oracle and reads through the OGR and GDAL libraries every conceivable raster and vector format, as well as OGC web services. QGIS has morphed from being its own desktop GIS to a shell that integrates open source GIS functions from GRASS, SAGA, gvSIG, and others. Like ArcGIS, it is relatively poor with regard to manipulating attribute data, which in both cases is better done outside the GIS.

While most core functions are written in C, QGIS offers both a Python console as well as a plethora of Python plugins or extensions provided by developers world-wide.

**VI.1.3 User-friendly Desktop Internet-oriented GIS (uDig)**

[uDig](http://udig.refractions.net/) is an open source spatial data viewer/editor, with special emphasis on the OGC standards for Internet GIS, the Web Map Server (WMS) and Web Feature Server (WFS) standards. The goal of uDig is to provide a complete Java solution for desktop GIS data access, editing, and viewing. There are standalone versions for Windows, Mac OS and Linux, however, for the purposes of Open Hazus, the application framework, built with Eclipse Rich Client (RCP) technology might be more promising. Their [gallery of sample applications](http://udig.refractions.net/gallery/) built with uDig illustrates implementations in hydrology, logistics, forest management, and the [Distant Early Warning System for Tsunamis](http://udig.refractions.net/gallery/dews/). uDig is also the underlying platform for DIVA GIS.

**VI.1.3.1 DIVA GIS**

[DIVA-GIS](http://www.diva-gis.org/) is a free GIS used for the analysis of geographic data, in particular point data on biodiversity, published under the GNU General Public license. The software was first designed for application to the study of wild potatoes in South America. DIVA-GIS was developed as a joint project by the International Potato Center in Peru, the International Plant Genetic Resources Institute, the Museum of Vertebrate Zoology at the University of California at Berkeley, the Secretariat of the Pacific Community, and the FAO. DIVA-GIS has a wide range of tools for evaluation of mapping and modeling of habitats. DIVA-GIS can process data in all standard GIS formats, including data from ESRI's ArcGIS programs. The program runs on Windows and OS X. DIVA raster files generated may be imported and exported into R or the modeling program [Maxent](https://biodiversityinformatics.amnh.org/open_source/maxent/).

**VI.1.3.2 uDig Rich Client Platform**

The previously discussed GRASS and QGIS programs were developed with end users in mind, i.e., a lot of effort has been expanded on provided a one-stop solution for the user’s GIS needs. uDig’s philosophy is different in that it acts more like a (Java-based) framework that allows [(experienced Java) developers](http://udig.refractions.net/developers/) to write their own GIS (like, for example, [JGRASS](http://udig.refractions.net/gallery/jgrass/)) based on the Eclipse platform.

**VI.1.4 gvSIG**

[gvSIG](http://www.gvsig.com/en/products/gvsig-desktop) is the third OSGeo desktop GIS application, published under the GNU/GPL license and used by thousands of users worldwide. gvSIG developed in parallel to QGIS and with a somewhat different emphasis on geospatial workflows. Its strengths are the geoprocessing component Sextante and its support for standards beyond the geospatial realm (e.g., databases, CAD, catalogues). gvSIG has a vibrant user and developer community, mostly in Europe and Latin America.

**VI.1.5 System for Automated Geoscientific Analysis (SAGA)**

[SAGA](https://sourceforge.net/projects/saga-gis/) started out as a set of programs the quantitative description of topographies (landform shapes). It has now become a serious competitor for the raster-based GRASS GIS. SAGA's overarching goal is to provide (geo-) scientists with an effective API for the implementation of geoscientific methods. Conceived of in the early 2000s and unencumbered by legacy code, SAGA's true strength lies in its architecture and very fast code.

SAGA is written in C++ and has an object oriented system design. It uses the cross-platform GUI library wxWidgets for user interface functionality. Because wxWidgets enables operating system independent software development, SAGA runs on MS-Windows as well as on Linux.

**VI.1.5.1 SAGA Capabilities**

One of the attractions of SAGA is its compact modular design that allows to run the software without installation even from flash drives. It has an unusually flexible command line interface that supports scripting in Python, Java, and R. There are more than 450 raster functions for geodata analysis (including a good amount of image processing and geostatistics) – but only minimal vector GIS support for clipping, buffering and raster to vector conversion.

**VI.1.6 Open JUMP (Java Unified Mapping Platform)**

Open JUMP lies somewhere between uDig and SAGA. It is the counter version of SAGA in that it concentrates on vector GIS capabilities, it mimics uDig in that it is written in and for Java, but it aims at end users providing a desktop solution for people who want a simple viewing platform for a variety of vector formats. The acronym JUMP stands for Java Unified Mapping Platform and suggests an emphasis on mapping rather than analysis. The mapping components are not particularly sophisticated (from a cartographer’s perspective) but will satisfy the needs of people who need SVG or web mapping output of data that sits in well-known spatial databases, CAD, Open Street Map (OSM) formats. For vector analytical operations, Open JUMP relies on the Victor Olaya’s Sextante that forms now the basis of the QGIS (geo-) processing framework.

**VI.2 Description of Open Source Web Mapping Applications**

The OSGeo Foundation supports some ten web mapping applications. Detailed descriptions can be found at <https://www.osgeo.org/projects>. The following notes concentrate on what might be useful in the context of this whitepaper, including non-OSGeo projects like Carto.

**VI.2.1 MapServer**

[MapServer](http://mapserver.org/) is one of the oldest web mapping engines around. Written in C, it runs on all major platforms, has some pretty advanced cartographic features and supports most OGC map-related standards. Although it can be customized to have a relatively light footprint, the full-fledged version with all bells and whistles is so feature-rich that it can be daunting to maintain. Best suited for developers who are not afraid of editing configuration files. MapServer can be used as a CGI or a plugin to various languages.

**VI.2.2 MapGuide**

Where MapServer is the well-established can-do-everything behemoth, MapGuide offers a very practical subset of functionalities that are easy to link with a server-side GIS. The cartography is not sophisticated but suitable for web-based applications. Unusual for a web *mapping* application, MapGuide supports a number of basic measurement and topology operations. It has a modern XML-based resource library that allows for a high degree of customization.

**VI.2.3 JavaScript libraries**

**VI.2.3.1 OpenLayers**

[OpenLayers](http://openlayers.org/) is a JavaScript library that allows for putting a dynamic map in any web page. It can display map tiles and markers loaded from any source. In contrast to the previously discussed applications, OpenLayers is not a full-fledged web mapping server software but an easy to use library that allows for light-weight web applications, leveraging WebGL and HTML5. This makes it eminently suitable for supporting mobile applications.

**VI.2.3.2 Leaflet**

Leaflet is a younger web mapping API that is designed to be lightweight, mobile-friendly, and easy to get started with. It places heavy emphasis on the use of tiled maps and client-side vector graphics drawn from sources such as GeoJSON. For basic maps that use these layer types, Leaflet is an excellent choice that has already endeared itself to many GIS developers. Impressive examples for spatio-temporal visualizations can be found at <http://apps.socib.es/Leaflet.TimeDimension/examples/>.

**VI.2.3.3 D3**

[D3.js](https://d3js.org/), an acronym for Data-Driven Documents, is a JavaScript data visualization library that is frequently used for charting but also contains many map examples. It binds data elements to the page’s document object model (DOM), allowing for interesting and flexible data animations and transitions. D3 is a nice option for composing a web app with interactive maps and charts. It also offers examples for using non-Mercator projections. Although D3 has originally been designed as a visualization tool, the library contains almost as many geospatial as visualization functions, especially once one considers shape and scale-related routines to fit neatly into either domains.

**VI.2.3.4 turf**

[Turf](http://turfjs.org/) is a JavaScript library for spatial analysis. It includes traditional spatial operations, helper functions for creating GeoJSON data, and data classification and statistics tools. Although developed by Mapbox, it is not a plugin but a standalone JavaScript library that works with any web mapping package.

**VI.2.3.5 Cesium**

[CesiumJS](https://cesiumjs.org/index.html) is a JavaScript library for world-class 3D globes and maps. It uses WebGL for hardware-accelerated graphics, and is cross-platform, cross-browser, and tuned for dynamic-data visualization. Cesium does not actually provide analytical functionality but makes for attractive fly-through visualizations of past and potential hazards. Melbourne's Intelligent Disaster Decision Support System ([IDDSS](https://www.unimelb.edu.au/cdmps/research/research-projects/iddss)) is a fine example for that.

**VI.2.3.6 kepler.gl**

[Kepler.gl](http://kepler.gl/#/) is a data-agnostic, high-performance web-based application for visual exploration of large-scale geolocation data sets. In the context of Open Hazus, it could be used for client-based exploratory analysis of massive (multi-million events) incidence data. The developers at Uber, the company where kepler.gl was developed, have been using it for dashboarding apps, and it is in this context, that this library could provide some of the functionality that has made Carto such a successful (easy to use) Web GIS.

**VI.2.4 deegree**

Java-based [degree](https://www.deegree.org/) differs from the two introduced web mapping applications in one important way: it has been written for spatial data infrastructures and includes OGC-compliant packages for cataloguing, data access, discovery and security. It handles all queryable properties of the Dublin Core and supports KVP as well as XML and SOAP requests. It is fully transactional, even for rich data models and supports a flexible mapping of GML application schemas (like the CityGML mentioned as a basis for the Open Hazus building inventory) to relational models. This makes degree uniquely qualified to become a component of the Open Hazus server – even if it is more commonly used in Europe rather than the United States.

**VI.2.5 GeoMOOSE**

[GeoMOOSE](http://www.geomoose.org/) is a browser based mapping framework for displaying distributed cartographic data. It is particularly useful for managing spatial and non-spatial data within county, city and municipal offices (from which GeoMoose originated). It extends the functionality of MapServer and OpenLayers to provide built in services, like drill-down identify operations for viewing and organising many layers, selection operations and dataset searches. GeoMOOSE is fast, performing well with hundreds of layers and/or services at a time. Data from multiple custodians can be maintained with different tools and on different schedules as each map layer has its own set of configuration files for publishing, symbols, templates as well as source data. The user interface is easily configurable, and additional services can be added through a modular architecture. GeoMOOSE allows for distributed maintenance amongst multiple owners – useful in Open Hazus’ multi user environment. Similar to degree (although not as sophisticated), it supports discovery and filtering from data catalogs.

**VI.2.6 Mapbender**

[Mapbender](https://mapbender.org/) is a content management system for geospatial data services and map applications. Mapbender is mostly written in JavaScript and based on the frameworks Symfony, JQuery and OpenLayers. Mapbender is platform independent and can be used on Windows, Mac and Linux. Mapbender has its own template for mobile applications, which has been optimized for use on smart phones and tablets. Administration of OGC Services in a Service repository. Individual configuration of Services in every application. Customization of applications to individual requirements via the web interface with a collection of customizable elements. The adaption of the designs can be done via customizable templates and CSS. Based on the integrated user management, individual applications, single functionality and services can be assigned to specific users or groups.

**VI.2.7 GeoServer**

[GeoServer](https://geoserver.org/) is an open source software server written in Java that allows users to share and edit geospatial data. It is the reference implementation of the OGC’s Web Feature Service (WFS) and Web Coverage Service (WCS) standards, as well as a high performance certified compliant Web Map Service (WMS).

Started in 2001, GeoServer has become the more modern competitor of MapServer. Those involved with GeoServer project also founded the GeoTools project (see VI.3.1), an open source GIS Java toolkit. Together with PostGIS (see section X.Y) and OpenLayers, and with significant support by the company [Boundless](https://boundlessgeo.com/), GeoServer has become the industry standard in web mapping software. Similar to degree, it features cataloging services (although not as comprehensive as those of degree) and web processing services (see also VI.2.9). GeoServer is comprehensive enough to require either a dedicated staff person or the kind of commercial managed services that allows Boundless to place this product in the public domain and still stay in business. [GeoServer is widely used in US federal](https://boundlessgeo.com/federal/), state, and larger local government agencies. GeoServer has just been ported to run on Java 11.

**VI.2.8 Carto**

Carto (formerly CartoDB) is a complete web-based GIS that is entirely based on dozens of open source packages and hence itself [available in source code on GitHub](https://github.com/CartoDB). Similar to GeoServer, the plethora of packages (as of December 2018, more than 300) that make up Carto is so big that a company, [Vizzuality](https://www.vizzuality.com/), makes its living by hosting customized Carto GIS server instances and developing specialized front and backends. Carto does an excellent job of bridging the gap between complex data analysis under the hood and intuitive, dash board-based user interfaces that are serving a wide range of stakeholders. Carto has been shifting its emphasis from that of a Web-GIS to a big geospatial data visualization platform.

**VI.2.9 PyWPS**

[PyWPS](http://pywps.org/) is not a classic web mapping application but a platform for web-based geoprocessing services – in this case based on Python. PyWPS implements the OGC web processing services specification and integrates with GRASS, GDAL, R, Fiona, RasterIO, MapServer, and QGIS. It is ideal for setting up, running and archiving all Open Hazus process models.

**VI.3 Description of Open Source Geospatial Libraries**

Geospatial libraries are low-level resources for software developers. They underlie many of the previously discussed web-based and desktop GIS applications. Here, we list only libraries that serve classic GIS tasks; others, like for example image processing and remote sensing-related libraries are beyond the scope of this whitepaper.

**VI.3.1 GeoTools**

A classic example for an open source geospatial library is [GeoTools](http://geotools.org/). GeoTools is a Java code library that provides standards compliant methods for the manipulation of geospatial data. It implements OGC specifications as they are developed. The OGC holds the copyright on the library and has made it available under the LGPL license. Based on the open source Java Topology Suite (JTS), GeoTools provides interfaces for key spatial concepts and data structures. It features a stateless, low memory renderer that is particularly useful in server-side environments.

**VI.3.2 GDAL/OGR**

There is no commercial or open source GIS product in active development anymore that does not rely on the geographic data abstraction library or [GDAL](https://gdal.org/). GDAL is a C++ translator library for 240+ raster and vector formats. Like GeoTools above, it is now managed and licensed by OGC. Its API serves C, C++, Python, Perl, C# and Java.

**VI.3.3 GEOS**

The Geometry Engine – Open Source library or [GEOS](https://trac.osgeo.org/geos) for short, is a C++ port of the Java Topology Suite. As such, it aims to contain the complete functionality of JTS in C++. This includes all the OGC Simple Features for SQL spatial predicate functions and spatial operators, as well as specific JTS enhanced topology functions.

**VI.3.4 GeoTrellis**

[GeoTrellis](https://geotrellis.io/) is a Scala library and framework that uses Apache Spark to work with raster data. It is released under the Apache 2 License. GeoTrellis reads, writes, and operates on raster data as fast as possible to transform user interaction with geospatial data by bringing the power of geospatial analysis to real time, interactive web applications.

GeoTrellis is a general framework for low-latency geospatial data processing developed using Scala and Akka. The goal of the project is to transform user interaction with geospatial data by bringing the power of geospatial analysis to real time, interactive web applications. It is complementary to other open source geospatial projects such as GeoServer, OpenLayers and PostGIS. GeoTrellis aims to:

* Create scalable, high performance geoprocessing web services;
* Create distributed geoprocessing services that can act on large data sets; and
* Parallelize geoprocessing operations to take full advantage of multi-core architectures

GeoTrellis is designed to help a developer create simple, standard REST services that return the results of geoprocessing models. In an application that relates to Open Hazus, the USACE Institute for Water Resources (IWR) is using the GeoTrellis framework for their Water Infrastructure Systems Data Manager to supports budget decisions based on multiple geographic criteria with real-time visual feedback as assumptions are interactively adjusted.

**VI.3.5 GeoWave**

[GeoWave](http://locationtech.github.io/geowave/index.html) is an open source library that leverages the scalability of a distributed key-value store for effective storage, retrieval, and analysis of massive geospatial datasets. Whereas Geo Trellis is more rster oriented, GeoWave adds multi-dimensional indexing capability to Apache Accumulo, HBase and Cassandra as well as Google's BigTable and Amazon DynamoDB, thereby supporting geographic objects and operators. The library comes with a plugin for GeoServer that allows geospatial data in Accumulo to be shared and visualized via OGC standard services.

GeoWave provides Map-Reduce input and output formats for distributed processing and analysis of geospatial data, as well as a PDAL plugin for interacting with point cloud data in Accumulo through the PDAL library. As such, GeoWave attempts to do for distributed key-value stores as PostGIS does for PostgreSQL (see also VII.1).

**VI.3.6 SpatioTemporal Asset Catalog**

The [SpatioTemporal Asset Catalog (STAC)](https://github.com/radiantearth/stac-spec) specification aims to standardize the way geospatial assets are exposed online and queried. A spatiotemporal asset is any file that represents information about the earth captured in a certain space and time. While the initial focus is primarily remotely-sensed imagery, the core is designed to be extensible to SAR, full motion video, point clouds, hyperspectral, LiDAR and derived data like NDVI, Digital Elevation Models, mosaics, etc.

The goal is for all major providers of imagery and other earth observation data to expose their data as SpatioTemporal Asset Catalogs, so that new code doesn't need to be written whenever a new JSON-based REST API comes out that makes its data available in a slightly different way. This will enable standard library components in many languages. STAC can also be implemented in a completely 'static' manner, enabling data publishers to expose their data by simply publishing linked JSON files online.

**VI.4 Description of Other Open Source Platforms**

HAZUS has entirely been based on GIS. This chapter of the whitepaper has so far been based on geospatial solutions that are closely tied to traditional GIS, even as the horizon expanded to open source desktop and web-based applications, as well as some of the lower-level libraries that these applications are based on. Two other aspects deserve consideration though. One is that awareness of geospatial functionality has spread beyond traditional GIS audiences, resulting in an increasing number of software packages that provide more than enough of the geospatial functionality needed for Open Hazus. The other is the fact that (Open) Hazus can be seen as an instance of a more generic family of software called spatial decision support systems (SDSS). There are multiple ways to implement an SDSS but increasingly, this is not on top of a GIS but like Repast Simphony discussed beneath, as a set of interlinked libraries that are embedded in general purpose development environments.

**VI.4.1 R**

[R](https://www.r-project.org/) is a language and environment for statistical computing and graphics. Readers of this Whitepaper might be surprised that R is considered a general enough platform to allow an Open Hazus to be built on top of it. But R has a huge developer community, including in all aspects of a spatial decision support system. There are very active communities for [geo-spatial](https://cran.r-project.org/web/views/Spatial.html), [operations research](https://cran.r-project.org/web/views/Optimization.html) and [machine learning](https://cran.r-project.org/web/views/MachineLearning.html) tools. Packages like [rattle()](https://cran.r-project.org/web/packages/rattle/index.html) illustrate how convenient it is to develop a workflow-based user interface. The main drawback to conceiving of R as an SDSS platform is the underdeveloped link to databases. There are plenty of packages that aim to provide links to DBMS, but they are slow and a de facto bottleneck. Calling R routines from a DBMS would be far more convincing.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

* an effective data handling and storage facility,
* a suite of operators for calculations on arrays, in particular matrices,
* a large, coherent, integrated collection of intermediate tools for data analysis,
* graphical facilities for data analysis and display either on-screen or on hardcopy, and
* a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term “environment” is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in R, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

R can be extended (easily) via [packages](https://cloud.r-project.org/). There are about 13,640 packages available on the Comprehensive R Archive Network ([CRAN](https://www.r-project.org/)) that extend the core functionality of R in, among others, the realms of Bayesian modeling, cluster analysis, environmetrics, machine learning, official statistics, spatio-temporal analysis, and web technologies

**VI.4.2 Repast Simphony**

Repast (Recursive Porous Agent Simulation Toolkit) is probably the most widely used free and open-source, cross-platform, agent-based modeling and simulation toolkit, developed at by Argonne National Laboratory and is now managed by the non-profit volunteer organization ROAD (Repast Organization for Architecture and Development). Repast provides a core collection of classes for the building and running of agent-based simulations and for the collection and display of data through tables, charts, and graphs. A particularly attractive feature of Repast is its ability to integrate GIS data directly into simulations.

Repast has been released in five versions supporting model development in various programming languages,

* Repast (Java based)
* RepastPy (based on the Python Scripting language)
* Repast.Net (implemented in C# but any .Net language can be used)
* RepastS (Repast Simphony, Java based, designed for use on workstations and small computing clusters)
* Repast High Performance Computing (an expert-focused C++-based modeling system designed for use on large computing clusters and supercomputers)

.. as well as built-in adaptive features, such as genetic algorithms and regression.

**VI.4.3 Eclipse**

[Eclipse](https://www.eclipse.org/) is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins for more than 25 languages. Plug-ins can be plugged-stopped dynamically.

From an Open Hazus perspective, it would be important to hide compiling, debugging, etc. components of the user interface in favor of the model definition, diagraming, testing, publishing, etc. tasks. The main reason to go through this effort is the universality of the tool. As Eclipse supports all the previously mentioned libraries, it is a convenient provisioning platform to manage all aspects of Open Hazus. The above mentioned Repast Simphony is a prime example for such an approach.

**VI.4.4 Jupyter**

A [Jupyter](https://jupyter.org/) notebook is an excellent means of capturing code, text, widgets, graphics, and other rich media in a computational narrative that distills data into insights. Today, notebook authors can share their notebooks for others to view and run in the Jupyter Notebook web application. Jupyter Notebooks are revolutionizing the way engineers and data scientists work together. Jupyter is built for writing and sharing code and text, within the context of a web page to build a "computational narrative that distills data into insights." (IBM 2015[[11]](#footnote-11))

While Jupyter's roots are in Python, it is now multi-lingual. Language support comes by way of modular kernels for more than 50 programming languages. The Toree kernel supports the Scala programming language and the Spark distributed computing platform (plus Python and R), simplifying the task of building large, data-intensive distributed applications.

Jupyter comes in three parts, a notebook frontend, a Jupyter server, and a kernel protocol. The frontend is a JavaScript web application that supports real-time dashboards. Jupyter can be easily extended. There are widgets for creating maps based on OpenStreetMap, and interactive data visualization with d3.js (see VI.2.3.3). The [Urban Data Science Toolkit](http://udst.squarespace.com/) is a fine example for an Open Hazus-like SDSS on Jupyter.

**VI.5 Decision Analysis Report**

Asses the various open source products and solutions available for Open Hazus GIS support to determine the set of tools and/or frameworks which provide the most functionality for desktop, web-based and mobile versions of Open Hazus.

[](file:///C:\Users\Jordan%20Burns\Documents\Projects\OpenHazus\OpenHazusDecisionAnalysis.xlsx)

**VI.5.2 Recommendation**

**VI.5.3 Rationale**

**VII. Database**

Open Hazus will require extensive database support both on the server and client side. As a large percentage of the data is geospatial in nature, it is important to have that aspect being explicitly supported by the database(s) of choice. In addition, especially the server-side database should have OLAP support to facilitate the mining of results databases. Finally, as the requirements for the inventory databases, depth grids, and possibly spatio-temporal features of dynamic hazard models keep increasing the storage and accessibility demands, non-relational options that still support geospatial needs should be considered either for future generations of Open Hazus or as a separate profile of the open solution set.

**VII.1 PostGIS**

[PostGIS](https://postgis.net/) is a spatial database extension for the PostgreSQL DBMS. It provides new types to PostgreSQL geometry, geography, raster, and topo-geometry and SQL/MM OGC Simple Feature SQL compliant functions for doing GIS work such as cadastral management, back-end for Web mapping services. It can also be used for network and road-routing using the complimentary [pgRouting](https://www.osgeo.org/projects/pgrouting/) PostgreSQL extension. Although there are spatial extensions to many other commercial and open source relational DBMS, PostGIS has become the de facto standard, definitely in the open source community but increasingly also in enterprise environments that have a well-established DBMS infrastructure in support of non-geospatial applications.

**VII.1.1 PostGIS Capabilities**

PostGIS is unique in the DBMS domain for the range of geospatial support it provides, including:

* Processing and analytic functions for both vector and raster data for splicing, dicing, morphing, reclassifying, and collecting/unioning with the power of SQL
* raster map algebra for fine-grained raster processing
* Spatial reprojection SQL callable functions for both vector and raster data
* Support for importing / exporting ESRI shapefile vector data via both command line and GUI packaged tools and support for more formats via other 3rd-party Open Source tools
* Packaged command-line for importing raster data from many standard formats: GeoTiff, NetCDF, PNG, JPG to name a few
* Rendering and importing vector data support functions for standard textual formats such as KML,GML, GeoJSON, GeoHash and WKT using SQL
* Rendering raster data in various standard formats GeoTIFF, PNG, JPG, NetCDF, to name a few using SQL
* Seamless raster/vector SQL callable functions for extrusion of pixel values by geometric region, running stats by region, clipping rasters by a geometry, and vectorizing rasters
* 3D object support, spatial index, and functions
* Network Topology support
* Packaged Tiger Loader / Geocoder/ Reverse Geocoder / utilizing US Census Tiger data

**VII.2 SpatiaLite**

Like PostGIS to Postgres, [SpatiaLite](http://www.gaia-gis.it/gaia-sins/) is a [SQLite](http://www.sqlite.org/) database engine with spatial functions added. Each SQLite database is simply a file that can be copied, compressed and ported between MS Windows, Linux, Mac OS, etc. The SpatiaLite extension enables SQLite to support spatial data conformant to OGC specifications. SpatiaLite is *not* intended for server-side use and should not be used for databases beyond a few GBytes. However, it works very well for desktop and mobile applications (including on Chrome OS and Android devices). In particular, SpatiaLite offers:

* Standard WKT and WKB formats
* SQL spatial functions such as AsText(), GeomFromText(), Area(), PointN() …
* OpenGis spatial analysis functions provided via GEOS, such as Overlaps(), Touches(), Union(), Buffer() …
* Full Spatial metadata in line with OpenGis specifications
* Numerous Geometry notations - EWKT, GML, KML, and GeoJSON
* Importing and exporting to shapefiles
* Coordinate reprojection via PROJ.4 and EPSG geodetic parameters dataset
* Locale charsets via GNU libiconv
* Spatial Index based on the SQLite’s RTree extension
* Access shapefiles as VIRTUAL TABLEs, enabling standard SQL queries on external shapefiles, without importing or converting them
* Access external CSV/TxtTab files or xls spreadsheets as VIRTUAL TABLEs
* Access XML documents, stored BLOB compressed binary objects, including syntactic “well formed” and XSF schema validation constrained checks. Specific support for ISO-Metadata, SLD/SE styles and SVG graphics.XML documents can be queried using standard XPath syntax.
* Query external WFS servers.
* Parse external DXF files (all versions) and store layers and geometries found.
* Generate and Export DXF files

**VII.3 rasdaman**

[rasdaman](http://www.rasdaman.org/), short for raster data manager, allows storing and querying massive multi-dimensional arrays, such as sensor, image, simulation, and statistics data appearing in domains like earth, space, and life science. This array analytics engine distinguishes itself by its flexibility, performance, and scalability. rasdaman can process arrays residing in file system directories as well as in databases. rasdaman is the first fully implemented, operationally used system with an array query language and optimized processing engine with unprecedented scalability. Known rasdaman databases exceed dozens of TB; [EarthServer](http://www.earthserver.eu/), is establishing intercontinental fusion of Petabyte datacubes. rasdaman integrates smoothly with R, OpenLayers, Leaflet, NASA WorldWind, GDAL, MapServer, or ESRI ArcGIS.

**VII.4 OLAP Databases**

Relational databases are good for traditional ETL tasks but rather limiting in the context of mining the results of ensembles of model runs. OLAP databases address the rapid access to analysis results question, allowing the interactive analysis of multidimensional data from multiple perspectives. OLAP hypercubes are dissections of the multidimensional database, similar to how an octree tessellates a 3D space. Each cube contains aggregated data related to elements along each of its dimensions. A typical user interface to interact with the data cubes is a Pivot table. The combination of all possible aggregations plus the base data contains the answers to every query that can be answered from the data.

**VII.4.1 Kylin**

Built on top of Hadoop/Spark [Apache Kylin](https://kylin.apache.org/) has a SQL interface, and can be used to carry out OLAP multidimensional analysis on Hadoop supporting extremely large datasets. The Kylin OLAP Engine is made up of a metadata engine, a query engine, a job engine and a storage engine. It also includes a REST Server to service client requests. The addition of support for RDBMS data sources means that data in formats such as Oracle, SQL Server and MySQL can be used within the analyses. Tools such as Apache Sqoop can also be used to export the data from RDBMS to HDFS to make it easier for Kylin to get the data and then build that into cubes. Kylin’s ODBC and JDBC drivers support tools such as Tableau for immediate graphical inspection.

**VII.4.1 Pinot with SpatialHadoop**

Another classic OLAP database is the column-oriented [Pinot](https://engineering.linkedin.com/teams/data/projects/pinot) datastore developed by LinkedIn. The data is stored in an HDFS (Hadoop distributed file system), which allows it to be linked with SpatialHadoop (2018). Hadoop’s Pig Latin in turn can be extended via user-defined functions, which users can write in Java, Python, JavaScript, Ruby or Groovy, and in the case of SpatialHadoop mimic PostGIS’s functionality. As a Java application, Pinot runs on all virtually operating systems.

[SpatialHadoop](http://spatialhadoop.cs.umn.edu/) is an extension to Hadoop that provides efficient processing of spatial data using MapReduce. It provides spatial data types to be used in MapReduce jobs including point, rectangle and polygon. It also adds low level spatial indexes in HDFS such as Grid file, R-tree and R+-tree. SpatialHadoop extends Hadoop in a similar way as PostGIS does with Postgres, providing spatial operations that are implemented as MapReduce jobs that access spatial indexes using the new components. Developers can implement myriad spatial operations (e.g., range query, k-nearest neighbor and spatial join) that run efficiently using spatial indexes.

**VII.4.2 OpenCube Toolkit**

The European OpenCube Toolkit (2018) is a fine example of how all this can be put together to process RDF data cubes. All user interactions are browser-based, including the Pivot table view and all OLAP operations. The OpenCube Toolkit executes R scripts, has interactive visualization widgets, and even a map interface in support of OLAP operations on the geo-spatial dimension.

**VII.5 Non-relational**

There is a little of an overlap with the previous section as some of the open source OLAP databases, when based on the HDFS, are also non-relational. Non-relational databases are typically built for specific data models (document, graph, key-value, etc.) and have very flexible schemas. They also scale very well. The tradeoff is that non-relational databases cannot guarantee the four hallmarks of Codd’s (1970) RDBMS, namely atomicity, consistency, isolation and durability (although RavenDB claims to be ACID). Excluded from this preview here are the two popular NoSQL databases from Amazon (DynamoDB) and Google (Firebase) because they do not qualify as free and open source (FOSS). The same holds for MongoDB, which is published under a so-called “server-side public license”, which does not fit the needs of the Open Hazus project.

**VII.5.1 CouchDB**

Apache’s [Cluster of Unreliable Commodity Hardware (Couch)](https://github.com/apache/couchdb) database features a document-oriented NoSQL architecture to store JSON data that can be queried with JavaScript. The main design goal was to have software that scales from mobile phones (it runs on Android and iOS) to cloud-based Big Data stores. CouchDB has drivers for over 15 programming languages. One of CouchDB’s strengths is the ability to synchronize two copies of the same database. Its replication feature supports both transient and persistent states, which might prove useful for multiple instances of Open Hazus' inventory databases. The replication protocol should address concerns of those 5% of Hazus users who require their own offline databases. CouchDB does not know of spatial data types or functions (other than the above radius around a point query) but it has the ability to create geospatial indices and to run radius queries using a number of distance measures around point locations.

**VII.5.2 Orient DB**

[OrientDB](https://github.com/orientechnologies/orientdb) is a schema-free, graph-based database, written in Java, with a SQL-like query language but no ability to performs classic relational joins. It has drivers for about a dozen popular programming languages and runs a choice of JavaScript or Java server-side scripts. As of 2016, OrientDB has a [spatial module](https://orientdb.com/database/spatial-module-orientdb-2-2/) that supports a number of vector geometries and a subset of SQL-MM functions (the only coordinate system supported is WGS84). Geospatial data can be im- and exported as CSV files with geometries in WKT. The original developer spun off a company OrientDB, which has an interesting Open Hazus-related [case study on their website](https://orientdb.com/database/traffic-management-white-paper/).

**VII.5.3 redis**

The [Remote Dictionary Server (redis)](https://redis.io/) is the most widely used NoSQL database implementing a distributed, *in-memory* key-value database with optional durability. Redis supports different kinds of abstract data structures, such as strings, lists, maps, sets, sorted sets, streams and spatial indexes with radius queries. Redis is written in ANSI C and hence works on most operating systems (Microsoft is maintaining the Windows version). The internal scripting language is Lua but there are drivers for some 35 programming languages. Redis does not know of spatial data types or functions (other than the above radius around a point query) but there is a [Geodis library on GitHub](https://github.com/EverythingMe/geodis) that provides geohashing to index locations. Redis has been benchmarked as the world’s fastest database but this comes at the price of being limited to in-memory operation, which precludes typical Open Hazus applications.

**VII.5.4 RavenDB**

[RavenDB](https://ravendb.net/), developed by a company that goes by the name of Hibernating Rhinos, is available on [GitHub in source code](https://github.com/ravendb) and precompiled for a number of different operating systems for small to medium sized applications. But to use, for example, SQL ETL one would have to purchase commercial licenses at not insignificant costs. Unusual for open source software, RavenDB is.net-based (yet runs on many operating systems and has drivers for six popular programming languages). What really sets RavenDB apart is the fact that despite being a NoSQL database it supports all ACID requirements and most important for Open Hazus, RavenDB has built-in geo-spatial indexing, spatial datatypes (WKT) and four spatial relationships. It supports server-side scripts, concurrency, and Hadoop. As the database itself has relatively small demands on hardware resources, it might serve well as a particular Open Hazus application profile for users that have streaming data demands.

**VII.5.5 Cassandra**

Originally developed in-house at Facebook, Cassandra is now an Apache NoSQL database that is geared to provide high availability across multiple datacenters. The obvious use case for Open Hazus is in case of a Katrina- or Sandy-like event that might cause regional web services to be unavailable. Cassandra’s own query language has drivers for a dozen common programming languages. Brahmin *et al.* (2016) developed a spatial extension for Cassandra. An example for its application can be found in the [tutorials for GeoMesa](https://www.geomesa.org/documentation/tutorials/geomesa-quickstart-cassandra.html) (see also VII.5.6).

**VII.5.6 GeoWave**

[GeoWave is a OSGeo library](https://www.osgeo.org/projects/geowave/) that connects the scalability of distributed computing frameworks and key-value stores with modern geospatial software to store, retrieve and analyze massive (i.e., terabyte) geospatial datasets. It utilizes distributed computing clusters and server-side fine grain filtering to execute interactive time and/or location specific queries on datasets containing billions of features.

GeoWave adds multi-dimensional indexing to Accumulo, HBase, BigTable and others. It indexes multidimensional data in a way that ensures values close together in multidimensional space are stored physically close together in the distributed datastore of a client's choice. GeoWave allows for easy feature extension and platform integration – bridging the gap between distributed technologies and minimizing the learning curve for developers. A GeoServer plugin allows geospatial data in a GeoWave datastore to be shared and visualized via OGC standard services. In addition, GeoWave provides Map-Reduce input and output formats for distributed processing and analysis of geospatial data.

**VII.6 Decision Analysis Report**

Asses the various open source products and solutions available for Open Hazus database support to determine the set of tools and/or frameworks which provide the most functionality for the server side of Open Hazus.

[](file:///C:\Users\Jordan%20Burns\Documents\Projects\OpenHazus\OpenHazusDecisionAnalysis.xlsx)

**VII.6.2 Recommendation**

**VII.6.3 Rationale**

.

**VIII. Analysis of Alternatives**

The purpose of this whitepaper is to define the functionality, external interfaces, performance, attributes, and design constraints of Open Hazus. It is based on a Product Requirements Assessment (PRA) focused on the major capabilities and features needed by its stakeholders. This document will not define additional capabilities but focuses on the provision of the capabilities as defined in the PRA. Because Open Hazus will be distributed for free by the federal government, other topics, such as pricing, competition analysis, marketing issues, are not relevant and are not addressed.

**VIII.1 Preferred Solution**

Multi-tiered access.

**VIII.1.1 Inventory Access**

Open Hazus users may include up to 50 variables in a single API query and make up to 500 queries per

**VIII.2 Security Considerations**

**VIII.3 Assumptions and Dependencies**

**IX. Implementation**

The purpose of this whitepaper is to define the functionality, external interfaces, performance, attributes, and design constraints of Open Hazus. It is based on a Product Requirements Assessment (PRA) focused on the major capabilities and features needed by its stakeholders. This document will not define additional capabilities but focuses on the provision of the capabilities as defined in the PRA. Because Open Hazus will be distributed for free by the federal government, other topics, such as pricing, competition analysis, marketing issues, are not relevant and are not addressed.

**IX.1 Open Hazus Server Access**

Multi-tiered access.

**IX.1.1 Inventory Access**

Open Hazus users may include up to 50 variables in a single API query and make up to 500 queries per IP address per day. More than 500 queries per IP address per day requires that users to register for an OHI (Open Hazus inventory) key. That key will be part of their data request URL string.

Stakeholders should keep in mind that all queries from a business or organization having multiple employees might employ a proxy service or firewall. This will make all of the users of that business or organization appear to have the same IP address. If multiple employees were making queries, the 500-query limit would be for the proxy server/firewall, not the individual user.

**IX.1.1.1 Registering for an OHI key**

Users will be directed to go to <https://www.hazus.fema.gov/developers> and click on the Request a KEY box. Upon filling the pop-up window form, they will then receive an email with their key code in the message.

**IX.1.2 Modeling access**

**X. Testing and Prototyping**

The purpose of this whitepaper is to define the functionality, external interfaces, performance, attributes, and design constraints of Open Hazus. It is based on a Product Requirements Assessment (PRA) focused on the major capabilities and features needed by its stakeholders. This document will not define additional capabilities but focuses on the provision of the capabilities as defined in the PRA. Because Open Hazus will be distributed for free by the federal government, other topics, such as pricing, competition analysis, marketing issues, are not relevant and are not addressed.

**X.1 Optimize Extract-by-Value**

**X.2 Microsoft Footprints as OpenHazus Foundational Inventory**

**X.3 Importing from the MIP, feasibility**

**X.4 Inventory coverage of flood products**

**X.5 Open Source Utility for Hazus Data Export and Reporting**

Hazus inputs and results are stored in backend SQL tables not intended for direct access by users. Exporting Hazus loss model results out of Hazus for further visualization or analysis using other technologies has historically been cumbersome. FEMA has distributed some ESRI-based scripts that automate parts of the export process for a more streamlined user experience, but these scripts have comparatively slow runtimes, require advanced ESRI licensing, and export results in traditional ESRI-based geospatial formats. SQL databases can be accessed and manipulated through a wide range of open source analytical libraries and the vast majority of Hazus model results are effectively managed and visualized through non-spatial processes. GIS scripting in general and ESRI tools in particular can therefore be bypassed in the development of a simple utility that extracts and summarizes Hazus model results in universally recognized text and image file formats.

A prototype utility for Hazus data extraction and summarizing has been developed for high-level earthquake model results. The utility employs open source Python libraries to extract several earthquake damage data points from Hazus SQL tables, summarize those data points across a variety of categories, and join them to a shapefile of study region tracts converted from SQL Spatial. The utility produces four csv files and one shapefile summarizing earthquake loss results and publishes those results to a public FTP site in an average of less than 10 seconds.

This prototype is being expanded to include flood, hurricane, and tsunami loss results and to provide basic data visualizations (spatial and graphical) as standard image outputs. If this utility can be effectively implemented as a user-friendly web service, it will serve as a proof of concept for GIS-independent results reporting in a web-based, open source version of Hazus.

**X.5 Streamlined Nationwide Inventory Updates using HIFLD**

The Homeland Infrastructure Foundation-Level Data program provides regularly updated, centrally managed nationwide inventory datasets similar to those used by Hazus. Hazus inventory datasets, however, are costly and inefficient for the Hazus Program to maintain and are therefore significantly outdated. Dynamic integration of Hazus with HIFLD layers would eliminate the duplicative burden to continuously update nationwide facility databases. Scripts and supporting data are being developed in order to assign attributes required by Hazus loss models to HIFLD layers so that legacy Hazus facility data can be updated and OpenHazus facility data can be automatically and dynamically updated going forward. A working partnership between HIFLD and the Hazus Team has been established so that each program can make the changes required to implement this dynamic data link.

**X.6 Open Source, Web-Based Flood Module**

The Oregon Department of Geology and Mineral Industries (DOGAMI) recently developed an ESRI ArcGIS-based toolbox that provides the functionality of a Hazus site-specific flood analysis without using Hazus software. An open source prototype of this ESRI toolbox was developed in order to test the hypothesis that an open source, web-based implementation of Hazus is effective and accomplishable. Open source versions were designed using traditional QGIS plugins, Python adaptations for QGIS, and GDAL analytical web services independent of desktop GIS technology. Runtimes and outputs were compared among all three prototypes and the original ESRI ArcGIS toolbox. The web-based GDAL analytical service produced site-specific flood damage results significantly faster than other implementations and provides data outputs as versatile and lightweight CSV files.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Software | Runtime (my computer) | Input Format | Output Format |
| 0 | ArcGIS | 84 seconds | .gdb, .mdb, .shp | .gdb |
| 1 | QGIS | 300 seconds | .mdb, .shp | .shp |
| 2 | QGIS and CSV | 40 seconds, 15 seconds \* | .mdb, .shp | .shp, .csv |
| 3 | CSV and GDAL | 4 seconds | .csv | .csv |

**A1 Definitions, Acronyms, and Abbreviations**

AAL Average annualized loss

AAR After action report

AEBM Advanced Engineering Building Model

BFE Basic Flood Elevation

BIT Building Import Model

ADCIRC [ADvanced CIRculation model](http://adcirc.org/) for oceanic, coastal and estuarine waters, developed at UNC

CAS Chemical Abstracts Service registry number

CAT Crisis Action Team

CDMS Comprehensive Data Management System

CDS Customer and Data Services, section of FIMA’s Risk Management Division

CERT Community Emergency Response Team

CEMP Comprehensive Emergency Management Plan

COG Continuity of Government

COOP Continuity of Operations Plan

CISM Critical Incident Stress Management

CIKR Critical Infrastructure and Key Resources

CNMS [Coordinated Needs Management Strategy](https://www.fema.gov/media-library-data/1521832299221-9e218ec1310c357befe493e534482673/CNMS_Technical_Reference_Feb_2018.pdf)

DEM Digital Elevation Model

DHS (United States) [Department of Homeland Security](https://www.dhs.gov/)

EF Essential Facilities

FEMA [Federal Emergency Management Agency](https://www.fema.gov/), a unit of DHS

FGDC [Federal Geographic Data Committee](https://www.fgdc.gov/)

FIMA [Federal Insurance and Mitigation Administration](https://www.fema.gov/what-mitigation/federal-insurance-mitigation-administration), a unit of FEMA

FIRM Flood Insurance Rate Map

FIS Flood Insurance Study

FIT Flood Information Tool

GBS General Building Stock

GBT General Building Type

GIS Geographic Information System

H\*WIND [Hurricane Surface Wind Database](http://storm.aoml.noaa.gov/hwind/)

H&H Hydrologic and Hydraulic (modelling studies)

HEC [Hydrologic Engineering Center](http://www.hec.usace.army.mil/) (of the U.S. Army Corps of Engineers)

HEC-RAS [HEC’s river Analysis System](http://www.hec.usace.army.mil/software/hec-ras/) that models the hydraulics of water flow through natural streams

HFT Hazard Factor Tables

HIFLD [Homeland Infrastructure Foundation-Level Data](https://hifld-geoplatform.opendata.arcgis.com/datasets)

HPLF High Potential Loss Facilities

HVA Hazard Vulnerability Assessment

HSEEP [Homeland Security Exercise and Evaluation Program](https://www.fema.gov/media-library/assets/documents/32326)

INCAST Inventory Collection and Survey Tool

LFD Letter of Final Determination

LOMR Letter of Map Revision

MAP Mapping, Assessment and Planning

MIP Mapping Information Platform (part of CDS Risk MAP)

MMI Modified Mercali Intensity

MSC Map Service Center; one of five units in CDS (see above)

NAVD North American Vertical Datum

NHC [National Hurricane Center](https://www.nhc.noaa.gov/)

NED National Elevation Dataset

NEHRP National Earthquake Hazards Reduction Program

NFIP National Flood Insurance Program

NHD National Hydrography Dataset (1:24,000)

NHRAP National Hazard Risk Assessment Program

NTHMP National Tsunami Hazard Mitigation Program

NWIRP National Wind Hazards Reduction Program

NWIRP National Windstorm Impact Reduction Program

NWM [National Water Model](http://water.noaa.gov/about/nwm)

OGC [Open Geospatial Consortium](http://www.opengeospatial.org/)

ORNL [Oak Ridge National Laboratory](https://www.ornl.gov/)

P4 Risk MAP Project Planning and Purchasing Portal

PCII Protected Critical Infrastructure Information

PESH Potential Earth Science Hazards

PGA Peak Ground Acceleration

PGD Permanent Ground Deformation

PGV Peak Ground Velocity

PII Personal Identifiable Information

PODs Points of Distribution

Risk MAP Risk Mapping, Assessment and Planning, a unit of FIMA

SFHA Special Flood Hazard Area

SLOSH Sea, Lake and Overland Surges from Hurricanes model developed by NOAA

SSI Sensitive Security information

SWAN [Simulating WAves Nearshore](https://www.tudelft.nl/en/ceg/about-the-faculty/departments/hydraulic-engineering/sections/environmental-fluid-mechanics/research/swan/) wave model (developed at TU Delft, NL)

TIGER [Topologically Integrated Geographic Encoding and Referencing system](https://www.census.gov/geo/maps-data/data/tiger.html)

UDF User-Defined Facility

UI/UX User Interface / User Experience

USGS [United States Geological Survey](https://www.usgs.gov/)

WKT Well-known Text, an ISO and OGC markup language standard for vector geometries

WSEL Water Surface Elevation

**A2 References**

ANSI/IEEE 1016-1998, “Recommended Practice for Software Design Descriptions” Institute of Electrical and Electronics Engineers, Inc., New York, 1998.

ANSI/IEEE 1016-1998, “Recommended Practice for Software Requirements Documents” Institute of Electrical and Electronics Engineers, Inc., New York, 1998.

ANSI/IEEE 829-1998, “Standards for Software Test Documentation”, Institute for Electrical and Electronics Engineers, Inc., New York, 1998.

Arundel, S.T., Archuleta, C.M., Phillips, L.A., Roche, B.L., and Constance, E.W., 2015, *1-meter digital elevation model specification*: U.S. Geological Survey Techniques and Methods, Book 11, chap. B7, 25 p. with appendixes, <http://dx.doi.org/10.3133/tm11B7>.

Barrington-Leigh C, & A Millard-Ball 2017. The world's user-generated road map is more than 80% complete. *PloS one*, **12**(8), e0180698. doi:10.1371/journal.pone.0180698

Basiri A, Jackson M, Amirian P, Amir Pourabdollah, Sester M, Winstanley A, Moore T & L Zhang 2016. Quality assessment of OpenStreetMap data using trajectory mining, *Geo-spatial Information Science*, **19**:1, 56-68, DOI: 10.1080/10095020.2016.1151213.

Brahmin M, Drira W, Filali F and N Hamdi 2016. Spatial data extension for Cassandra NoSQL database. *Journal of Big Data*, **3**:11, DOI: 10.1186/s40537-016-0045-4.

Breakstone C and T Anderson 2017. Census Data API User Guide, version 1.5. Suitland, MD: The US Census Bureau.

Brzev S, Scawthorn C, Charleson AW, Allen L, Greene M, Jaiswal K and V Silva 2013. *GEM Building Taxonomy Version 2.0*, GEM Technical Report 2013-02 V1.0.0, 188 pp., GEM Foundation, Pavia, Italy, doi: 10.13117/GEM.EXP-MOD.TR2013.02.

Codd, E 1970. A relational model of data for large shared data banks. *Communications of the ACM*, **13**(6): -377-387.

Department of Homeland Security (DHS) Strategic Plan (2014 – 2018)

Federal Insurance and Mitigation Administration (FIMA) Strategic Plan (2012 – 2014)

FEMA Administrator’s Intent (2015 – 2019)

FEMA National Advisory Council (NAC) Recommendations (2016)

FEMA Strategic Plan (2014 – 2018)

Lloyd C, Sorichetta A and A Tatem 2017. High resolution global gridded data for use in population studies. *Scientific Data*, **4**: 170001. <https://www.nature.com/articles/sdata20171.ris>.

Natural Hazard Risk Assessment Program (NHRAP) Strategic Plan

NWIRP Year. Title.

Presidential Policy Directive 8: National Preparedness (PPD-8)

Risk Mapping, Assessments, Planning (Risk MAP) Program, FIMA Multi-Year Plan: Fiscal Years (2010 – 2014).

Sehra S, Singh J and H Rai 2014. A Systematic Study of OpenStreetMap data Quality Assessment. *Proceedings of the Eleventh International Conference on Information Technology*: New Generations. Las Vegas, NV. Berlin (DE): FI-STAR Consortium. DOI 10.1109/ITNG.2014.115.

Technical Mapping Advisory Council (TMAC) Program Vision and Goals (December 2015)

TMAC Annual Report (December 2015)

TMAC Future Conditions Report (December 2015)

UNECE 2017. *UN Recommendations on the Transport of Dangerous Goods. Model Regulations* (Twentieth ed.), New York and Geneva: United Nations, 2017, ISBN 978-92-1-139120-6, ST/SG/AC.10/1/Rev.20.

1. The coastal option is only enabled if the user previously ran the Hurricane model [↑](#footnote-ref-1)
2. Li Y and B Ellingwood 2009. “Framework for multihazard risk assessment and mitigation for wood-frame residential construction.” *J. Struct. Eng.*, 10.1061/(ASCE)0733-9445(2009), **135**:2(159), 159–168. [↑](#footnote-ref-2)
3. Sehra S, Singh J and H Rai 2014. A Systematic Study of OpenStreetMap data Quality Assessment. *Proceedings of the Eleventh International Conference on Information Technology: New Generations*. Las Vegas, NV. Berlin (DE): FI-STAR Consortium. DOI 10.1109/ITNG.2014.115.

   Basiri A, Jackson M, Amirian P, Amir Pourabdollah, Sester M, Winstanley A, Moore T & L Zhang 2016. Quality assessment of OpenStreetMap data using trajectory mining, *Geo-spatial Information Science*, **19**:1, 56-68, DOI: 10.1080/10095020.2016.1151213.

   Barrington-Leigh C, & A Millard-Ball 2017. The world's user-generated road map is more than 80% complete. *PloS one*, **12**(8), e0180698. doi:10.1371/journal.pone.0180698 [↑](#footnote-ref-3)
4. https://github.com/Microsoft/USBuildingFootprints [↑](#footnote-ref-4)
5. December 2018 [↑](#footnote-ref-5)
6. https://www.usgs.gov/news/fy19-us-geological-survey-broad-agency-announcement-3d-elevation-program-released [↑](#footnote-ref-6)
7. Popularly known as geoprocessing models in ESRI parlance but now in need to be generalized. [↑](#footnote-ref-7)
8. The World Bank, for instance, is hosting a [repository of links](https://github.com/WorldBank-Transport/GTFS-Training-Materials/wiki/Link-repository-for-international-GTFS-training-materials) to open source applications that support GTFS. [↑](#footnote-ref-8)
9. UN Recommendations on the Transport of Dangerous Goods. Model Regulations (Twentieth ed.), New York and Geneva: United Nations, 2017, ISBN 978-92-1-139120-6, ST/SG/AC.10/1/Rev.20 [↑](#footnote-ref-9)
10. Not for lack of detail to the methodology itself but for the scope of the calculation, which is a function of building damage and guestimates about their occupancy based on the different functions they serve and the time of the event. [↑](#footnote-ref-10)
11. <http://blog.ibmjstart.net/2015/08/22/dynamic-dashboards-from-jupyter-notebooks/> [↑](#footnote-ref-11)