Architectural Overview of MAEviz – HAZTURK

AMR ELNASHAI1, SHAWN HAMPTON2, JONG SUNG LEE2, TERRY McLaren2, JAMES D. MYERS2, CHRIS NAVARRO2, BILL SPENCER1, and NATHAN TOLBERT2

1Department of Civil and Environmental Engineering, University of Illinois, Urbana-Champaign, Urbana, Illinois, USA
2National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign, Urbana, Illinois, USA

MAEviz is a broadly extensible, open source platform for earthquake hazard risk management. MAEviz is a model cyberenvironment that provides practical capabilities for researchers through decision-makers to model earthquake events, develop risk reduction strategies, and implement mitigation plans to minimize the impact of earthquake disasters while also providing a pathway for researchers to quickly add new algorithms and data to assure that decisions are based on state-of-the-art engineering understanding. While MAEviz is capable of interacting with remote data and computational sources, it is also fully capable of running analyses locally so research scientists and decision-makers can generate information when a catastrophic event occurs and provide first-responders result information. This article describes MAEviz’s overall layered architecture, its foundation in the widely used Eclipse Rich Client Platform (RCP), and use of open-source middleware and geographic information system (GIS) components. MAEviz’s data management capabilities and workflow-oriented execution model are also discussed with an emphasis on detailing MAEviz’s capability to incorporate new data types and new analysis modules.

Keywords Consequence-Based Risk Management; Eclipse Rich Client Platform; NCSA GIS Extensible Platform; Risk Analysis; Risk Mitigation

1. Introduction

MAEviz has been developed as a new application enabling policy-makers and decision-makers to develop risk reduction strategies and implement mitigation actions to minimize the impact of earthquake disasters. MAEviz is also an extensible platform that enables the earthquake risk management community to quickly bring new data and modeling capabilities into practice. Risk management professionals are familiar with identifying vulnerabilities, assessing loss reduction strategies, guiding resource allocation before disasters, identifying vulnerable areas during disasters, guiding recovery efforts, and providing information to decision-makers throughout the process. Assuring that the science and engineering behind such forecasting of damage probability of buildings, bridges, pipelines, and other inventory items from anticipated events is state-of-the-art is therefore critical to minimizing the impact of earthquakes; in addition, it can save lives, reduce losses to economic resources, and lead to more stable and secure communities.

Many risk assessment tools and platforms exist today; however, most lack the flexibility to easily add new algorithms or extend their base features due to a combination of architecture
and closed-source licensing policies. Such software does not allow the community to actively contribute new algorithms and capabilities and allow the software to evolve with the advancements of science. Further more, software licensing fees can make the packages unaffordable for many members of the community, particularly in developing countries.

MAEviz incorporates many of the design concepts and capabilities motivated by National Center for Supercomputing Applications’ (NCSA) efforts to develop “Cyberenvironments” that span scientific disciplines and that can rapidly evolve to incorporate new research results [Myers and Dunning, 2006]. Specifically, MAEviz incorporates content management and workflow concepts that support dynamic Cyberenvironments [Myers and McGrath, 2007].

MAEviz’s reliance on an open-source business model, combined with its modular, extensible architecture address many of the issues that limit the utility of other analysis tools, and provide a potentially revolutionary capability to enable sophisticated risk analysis around the globe.

2. Background

MAEviz’s Earthquake Loss Assessment framework is based on the Eclipse Rich Client Platform [RCP, 2007] which has several notable advantages, including its strong industry backing, support for multiple platforms, support for Native-OS integration, and software extensibility through a dynamic ‘plug-in’ based architecture. The plug-in architecture of Eclipse provides extension points where new features can be dynamically added to the application, and communities can choose which feature plug-ins they want included in their tools. This extensibility is a primary reason for choosing Eclipse as the basis for the MAEviz framework. Furthermore, use of the Eclipse RCP, whose development is sponsored and used by industry leaders, such as IBM, Motorola, CA, Sybase, Cisco Systems Inc., and NASA, also provides MAEviz with an immense developer and user base that will continue to modify and extend the underlying Eclipse platform to make it robust and keep its technology current. This developer base has provided hundreds of add-ins to the framework that MAEviz can take advantage of as it evolves and grows, such as the Business Intelligence and Reporting Tools (BIRT), the Eclipse Communications Framework (ECF), and the Graphical Editor Framework (GEF). Eclipse is available for all of the major operating systems including Linux, Windows, Mac OSX, and UNIX.

Early in the design phase of the new framework, it became apparent that the Geographic Information System (GIS) functionality should be separate from the MAEviz plug-ins so that other applications could take advantage of the core GIS plug-ins. This gives MAEviz both horizontal and vertical extensibility. That is, users can extend the GIS core by adding additional features or they can extend MAEviz with additional earthquake analysis and science features. The next few sections describe in detail the Eclipse RCP, the NCSA GIS application container, the MAEviz plug-ins, and how to extend the MAEviz framework.

3. RCP – What is the Rich Client Platform

This section provides a brief overview of the Eclipse RCP. (A full introduction to Eclipse RCP is beyond the scope of this article, and readers are encouraged to see http://www.eclipse.org/rcp for more information.) The Eclipse RCP provides a base set of features (or plug-ins) that can be used to build new applications as shown in the architecture diagram in Fig. 1.

A plug-in is a component that provides a set of functionality to the RCP core or workbench. Each plug-in can provide stand-alone capabilities since every plug-in has its own class loader, which allows it to be dynamically instantiated on an as needed basis. The
plug-in may also define extension points which define how the plug-in code can be extended.
The Standard Widget Toolkit (SWT) is the User Interface (UI) toolkit that gives developers a full suite of UI features to build applications and includes widgets to create menu bars, toolbars, perspectives, views and editors, preferences, and removable windows. At runtime, the toolkit uses the native widgets for each operating system so developers do not have to deal with the details of cross platform look and feel issues. JFace is a library that provides a layer of abstraction to SWT providing a cleaner Application Programming Interface (API) to the UI code. The workbench, as shown in Fig. 1, is essentially a collection of these widgets. Underneath the workbench is the Eclipse Runtime which is formed by the Open Service Gateway initiative [OSGI, 2007] framework and provides the capability of dynamically loading plug-ins. This provides the smallest possible runtime memory footprint because only the plug-ins being used are loaded into memory.

4. NCSA GIS – A Rich Client Application

The set of NCSA GIS plug-ins are built upon the RCP core plug-ins and provide a platform for data management, visualization, and analysis execution, as shown in Fig. 2.

The data management features provide data typing, ingestion, access, and provenance tracking [Futrelle, 2006]. Each dataset must have an associated type, which makes an association with a default data schema and provides MAEviz information about its attributes, a process to ingest the data into the system and the ability to track provenance data by associating metadata with the dataset. The visualization system supports both 2D and 3D views, with the ability to add additional 2D/3D views through the extension point system. A few features supported by the visualization system are zoom, selection, and highlighting for data synchronization. For example, a selection in the attribute table of geospatial data will highlight the corresponding geospatial feature on the map. This can be useful when searching specific data in large datasets. NCSA GIS has also incorporated middleware from the Linked Environments for Atmospheric Discovery project (LEAD; Alameda et al., 2007) to provide support for local multi-threaded execution of analyses and, in future versions, for remote execution on supercomputing computer resources.
A visual dataflow system is available in Version 2.3 which will allow users to see individual analyses and their respective dependencies as well as state information. The state information will show users which analyses are ready to execute, which require their attention, and the current status of running analyses. In addition to these features, NCSA GIS accommodates geospatial functionality through a set of Open Geospatial Consortium [OGC, 2007] compliant open source libraries called Geotools [Geotools, 2007]. The Geotools library provides support for many GIS vector formats including ESRI Shapefile, GML, WFS, PostGIS, Oracle Spatial, ArcSDE, MySQL, GeoMedia, Tiger, VPF, and MIF. It also gives support for several raster formats including ESRI ArcInfo ASCII Grid Format, GRASS ASCII Grid Format, geo-referenced image format, and WMS. Some of the other notable features of NCSA GIS include: 2D rendering and styling, data aggregation, layer ordering, attribute filtering, grid overlays, and coordinate transformation.

The general architecture and dataflow of MAEviz is shown in Fig. 3. This pictorial overview starts with the data sources in the upper left, continues with the different functional classes used in managing data sets and collections of data in scenarios, and ends with the various views to the data in the lower left.

MAEviz is capable of interacting with multiple local or remote data sources and repositories. Currently, MAEviz supports access to the local file system, LAN and WAN network drives and the Scientific Annotation Middleware [SAM, 2007], a web-based content management repository [WebDAV, 2007]. Each data repository type is extended from the BaseRepository class which provides MAEviz a common baseline, and each repository plug-in defines how MAEviz connects to the repository. The BaseRepository class also provides the interface for retrieving and writing datasets to the repository. The box labeled DB repository highlighted in blue in Fig. 3 shows future data store options that can be supported.

The Registry class provides a local cache capability and quick reference to different data and analysis objects contained in a scenario. The AnalysisManager is the base class where all analyses are registered; developers can add a new analysis by registering their new plug-in with this extension point. The ScenarioManager class is responsible for
keeping track of all objects associated with a particular scenario in a user’s workspace. A user’s workspace can support multiple scenarios simultaneously, and each scenario is independent. The scenario contains all of the datasets associated with a user’s project as well as any settings applied to the datasets (e.g., color, visibility, etc.) and any datasets that can be visualized. When a user creates a new scenario through the workbench, they are prompted to define a region of interest, make an analysis selection, and identify the inventory data. The AnalysisManager then attempts to populate selected analysis fields with the appropriate data types and default input parameters, and in some cases, the user will need to provide parameter inputs. If a dataset for an analysis field (e.g., bridges) is not available in the user’s scenario, then the user has the option to search all registered repositories to find an appropriate dataset, or to create a new dataset using an analysis. If the user chooses to create a new dataset, the AnalysisManager will first execute the sub-analysis necessary to create the dataset, and then it will use the result as an input into the original analysis.

When an analysis has finished executing, the generated datasets are returned to the local cache (managed by the Registry) and the Scenario. Now the dataset is available to view in various forms. For example, the data can be viewed in a tabular format which supports common spreadsheet capabilities like sorting and filtering. If the dataset can be rendered, MAEviz will create a visualization view which supports both 2D and 3D
visualizations. MAEviz currently uses GeoTools to render the 2D views and the Visualization Toolkit (VTK) libraries to render 3D views; however, the visualization view is an extension point and can support other visualization capabilities and libraries.

5. MAEviz – An NCSA GIS Application

MAEviz is a data-centric network-aware application whose primary goal is to analyze and visualize data from potentially independent sources. Users create scenarios which define the associations between data, analyses, results, and various views into the data. MAEviz adds to the base NCSA GIS core through plug-ins that implement capabilities specific to earthquake science, as shown in Fig. 4.

The common data extensions in MAEviz include: hazards, buildings, bridges, and pipelines. By clearly defining these data types, MAEviz knows which seismic analyses can be performed with each dataset. For example, in building damage analysis, MAEviz defines a standard data schema specify attributes such as building type, age, area, etc. The user can access this schema through an import wizard and make associations between their data set, which may have different attribute names, and the MAEviz schema. This data import process provides a mechanism to create MAEviz compatible data sets on the fly by mapping the user’s data to the MAEviz standard data types. MAEviz will also store and display any additional data which is associated with the import data set; this provides a dynamic way to meet the data needs of new analysis modules and extend the default schema. By associating a dataset to a data type, MAEviz makes the data context sensitive to the different analyses. MAEviz, whose scientific functionality has been reported in the MAEviz Overview [Elnashai, et al., 2007], currently supports 36 analyses, including structural damage analyses for buildings, bridges, and pipelines, bridge functionality, bridge repair cost, building direct economic damage, fiscal impact, etc.

![MAEviz Architecture Diagram](image-url)

**FIGURE 4** MAEviz architecture diagram.
6. Extending MAEviz

The most common extension points in MAEviz support new data types and analysis modules. Adding a new data type requires the user to extend the ncsa.gis.gisSchemas extension point and create an xml-based schema description file. The schema description file specifies the data attributes and the data type (e.g., string, double, integer). Two common types of datasets in MAEviz are feature datasets (e.g., bridges and buildings) and raster datasets (e.g., hazard map, slope map). Importing these datasets with schemas allows MAEviz to provide the appropriate filtering options, ensure valid field values, and limit the allowable parameters in the UI fields, which are tied to specific input type. For example, a bridge damage analysis will not allow a building dataset as an input because it is the wrong type. Data types also allow the Registry to search for datasets that can be applied to a UI field since the field is aware of valid inputs. Adding a new dataset type to the framework has only two requirements: an XML based schema description file and registration of that file with the ncsa.gis.gisSchema extension point. Once registered, the new schema will be available in the data ingestion menu. This data type extension point is important because it allows users to import new data types that are not currently supported by the framework and then use them in existing or new analyses.

Adding new analyses to the framework is similar to adding a new dataset; however, it requires writing code to execute the analysis. Each analysis must register with MAEviz through the ncsa.analysis.newAnalyses extension point. In addition, an xml file describing the inputs, outputs, and runtime requirements must be created so the framework can generate the appropriate UI fields and divide the computational tasks appropriately. In addition, a Java-based class must be created to define the logic of the algorithm being implemented, along with the appropriate access methods to provide the inputs from the UI to the analysis task.

7. Conclusion

MAEviz was created as an extensible environment using state-of-the-art open source technologies and represents the next generation of software design and development in earthquake hazard risk management. The architecture was built to meet the needs of the earthquake engineering and risk assessment communities as well as other science disciplines in need of a GIS-capable platform. In addition, MAEviz provides a well-defined mechanism to upgrade data and analyses so its users can maintain pace with the latest science breakthroughs and theories. The users of MAEviz have a powerful tool with practical capabilities for researchers through decision-makers to model earthquake events, develop risk reduction strategies, and implement mitigation plans in order to minimize the impact of earthquake disasters while also providing a pathway for researchers to quickly add new algorithms and new data to assure that decisions are based on state-of-the-art engineering understanding.

References


