

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Earthquake Record Selection for Dynamic Structural Analysis: an Application with a Web Interface.

Marília Natália Seabra Areal Gonçalves

Report of Dissertation

Master in Informatics and Computing Engineering

Supervisor: João António Correia Lopes (Auxiliar Professor)

Second Supervisor: José Miguel de Freitas Castro (Auxiliar Professor)

29th June, 2009

Earthquake Record Selection for Dynamic Structural Analysis: an Application with a Web Interface.

Marília Natália Seabra Areal Gonçalves

Report of Dissertation

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Rosaldo José Fernandes Rossetti (Auxiliar Professor)

External Examiner: José Carlos Leite Ramalho (Auxiliar Professor)

Internal Examiner: João António Correia Lopes (Auxiliar Professor)

28th July, 2008

Abstract

In the field of seismic engineering it is common to employ dynamic analysis techniques to predict, to a certain degree of accuracy, the behaviour of a structure in the event of an earthquake. The American and European regulations specify two different types of seismic response spectra that are used for the seismic design and/or assessment for simple structures.

When more complex structures are analysed, the regulations specify that more advanced analysis should be performed which consists of assessing the response of a given structure to the action of a given set of earthquake records. According to the regulations these records should be compatible with the seismic response spectrum representing the earthquake hazard of the area where the structure will be located.

SeIEQ, an application developed for earthquake record selection, aims to help seismic engineers in finding adequate records for dynamic analysis. It allows the users to easily filter through thousands of real records, and order them by adequacy using a root-mean-square formula to compare them with the regulation-defined spectra.

SeIEQ has been developed for the Web, using a 3-tier client-server architecture. It does not force the users to have any specific software installed (a simple Web browser is enough to access the application), and it is easily maintainable and extensible (since the business rules and database are kept server-side).

It is hoped that SeIEQ will serve as a proof-of-concept for seismic engineers, displaying the capabilities of the Web for applications shared by several users.

Resumo

Na área de Engenharia Sísmica é comum utilizarem-se técnicas de análise dinâmica para prever, com um determinado grau de confiança, a resposta de uma estrutura a um sismo. As normas americanas e europeias definem um conjunto de espectros de resposta que são usados para estes cálculos quando se trata de estruturas bastante simples.

Quando é necessário analisar estruturas mais complexas, as normas especificam que devem ser efectuadas análises mais avançadas que consistem na avaliação da resposta da estrutura à acção de um conjunto de sinais sísmicos.

SeEQ, uma aplicação desenvolvida para pesquisa de registos sísmicos, pretende ajudar os engenheiros sísmicos a encontrar registos para análise dinâmica de estruturas. Esta aplicação permite aos seus utilizadores filtrar milhares de registos reais e ordená-los por uma função de valor eficaz que quantifica a diferença dos mesmos com o espectro definido pelas normas.

SeEQ foi desenvolvido para a Web e usa uma arquitectura cliente-servidor de três camadas. A aplicação não exige qualquer software específico (um navegador Web é suficiente para aceder a todas as funcionalidades da aplicação) e é facilmente alterável e extensível (dado que as regras de negócio e a base de dados são mantidas do lado do servidor).

Espera-se que a aplicação SeEQ sirva como uma prova de conceito para engenheiros sísmicos, demonstrando as capacidades da Web para aplicações partilhadas por vários utilizadores.

Acknowledgements

There are several people without whom it'd have been very hard to get so far and write something as scary as a whole dissertation.

I would like to thank both my supervisors, João Correia Lopes and José Miguel Castro, for their patience and willingness to help me during the course of the work, as well as for their constructive, detailed criticism;

My friends for their incredible support when words just wouldn't come, for the encouragement given during those hard days of writing, as well as the necessary fun moments we've shared during this semester that kept me going;

My boyfriend, for his support, patience and suggestions, as well as his willingness to help me when things weren't looking so well;

And my family, for bringing me to where I am.

Marília Gonçalves

*“In all science, error precedes the truth,
and it is better it should go first than last.”*

Hugh Walpole

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Objectives	2
1.4	Structure of the Dissertation	2
2	Web Application for Earthquake Record Selection	5
2.1	The Problem	5
2.1.1	Domain Concepts	6
2.1.2	Goodness-of-fit Calculations	6
2.2	Related Work	9
2.2.1	Structural Analysis Software	9
2.2.2	Matlab	9
2.2.3	REXEL	9
2.3	Web Technologies	10
2.3.1	Model-view-controller Architectural Pattern	10
2.3.2	ASP.NET	10
2.3.3	PHP	11
2.3.4	Ruby on Rails	12
2.3.5	Comparison and Conclusions	12
2.4	Relational Database Systems	13
2.4.1	MySQL	13
2.4.2	PostgreSQL	14
2.4.3	Other Database Systems	14
2.4.4	Comparison and Conclusions	14
2.5	Conclusions	14
3	SeEQ Specification	17
3.1	Introduction	17
3.2	Seismic Engineering Requirements	17
3.2.1	Filtering of Data	18
3.2.2	Calculation of Spectra	18
3.2.3	Goodness-of-fit Calculations	19
3.2.4	Exporting Records	19
3.3	System Requirements	19
3.3.1	Non-functional Requirements	19
3.3.2	Functional Requirements	20

CONTENTS

3.4	Scenarios of Usage	21
3.5	Conclusions	24
4	Design & Implementation	25
4.1	Architecture Design	25
4.1.1	Web Server – Apache	25
4.1.2	Protocols	26
4.2	Application Data & Data Model	26
4.2.1	Denormalisation	27
4.3	Programming Languages and Applications	29
4.3.1	Programming Languages	29
4.3.2	Markup Languages and Style	30
4.3.3	Support Applications	30
4.4	Calculations	31
4.5	Conclusions	31
5	Evaluation and Case Study	33
5.1	Evaluation	33
5.1.1	Functional and Non-functional System Requirements	33
5.1.2	Practical Use	33
5.2	Exploitation	34
5.2.1	Filtering	34
5.2.2	Insertion of D_{rms} Information	34
5.2.3	Viewing Results	34
5.2.4	Ordering Results	37
5.2.5	Calculation of D_{rms}	37
6	Conclusions and Future Work	39
6.1	Achievements	39
6.2	Future Work	39
6.2.1	Generation of Spectra	40
6.2.2	Filtering by Ground Type/Site Class	40
6.2.3	Geospatial Search	40
6.2.4	Generate PDF Report	40
	References	41
A	American Regulations	43
B	European Regulations	45

List of Figures

2.1	Domain Object Model	7
2.2	Shape of the European Elastic Response Spectrum	8
2.3	Model-view-controller Pattern Behaviour	11
2.4	Comparison of Programming Languages	13
3.1	Use Case 1 – Filtering	22
3.2	Use Case 2 – Ordering	23
3.3	Use Case 3 – Exporting	24
4.1	SeIEQ Architecture	26
4.2	Normalised Database	28
4.3	Denormalised Database	28
5.1	Selection of Filters	34
5.2	Insertion of Filter Information	35
5.3	Insertion of D_{rms} Information	35
5.4	Viewing of Results	36
5.5	Viewing an Individual Record	36
5.6	Results Ordered by Magnitude	37
5.7	D_{rms} calculation	38

LIST OF FIGURES

List of Tables

A.1	Site Coefficient F_a	44
A.2	Site Coefficient F_v	44
B.1	Type 1 Elastic Response Spectra Parameters	46
B.2	Type 2 Elastic Response Spectra Parameters	46

LIST OF TABLES

Abbreviations

Computer science

API	Application Programming Interface
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
DOM	Document Object Model
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
MVC	Model-view-controller architecture
PgSQL	Postgres Structured Query Language
PHP	PHP: Hypertext Preprocessor
RoR	Ruby on Rails
SQL	Structured Query Language
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
WWW	<i>World Wide Web</i>

Seismic Engineering

D_{rms}	Root-mean-square difference
-----------	-----------------------------

ABBREVIATIONS

Chapter 1

Introduction

This chapter describes the context, motivation, and objectives for this project. It finishes with a brief description of the structure of this dissertation.

1.1 Context

Seismic analysis has become increasingly important when planning and designing a new building or structure. This analysis aims to guarantee a degree of structural stability in the event of an earthquake, with several required levels of performance; for example, the stability and resistance requirements of a hospital to seismic waves are distinctly more important than that of a warehouse.

Choosing the right seismic records for this analysis is an important part of the process. Two different criteria can be used for this purpose. The first considers site conditions (geophysical criteria¹), which leads to record selection based on criteria such as magnitude of the earthquake or distance to the fault, or typically a combination of the two. The second is based on strong-motion parameters² – the seismic records are picked according to how well they conform to the theoretical model established for that area.

Databases for seismic records typically hold thousands of records. Because of this, and regardless of the criteria chosen for selection or shortlisting of these seismic records, there is a clear need to automate the process so that the best records are chosen over other less useful possibilities. Particularly when deciding based on strong-motion parameters, there are also several calculations to be made (so as to order the records by how similar they are to the target) which are not time-efficient if the whole process is to be carried out by hand.

¹Quantitative physical properties of the Earth.

²Strong earthquake shaking that occurs close to a causative fault.

A full description of this process can be found under section 2.1.

1.2 Motivation

The tools currently in use by seismic engineers do not allow for an ideal productivity level. There is certainly room for improvement where software tools can be developed to aid engineers. Because of this, a joint effort between civil and informatics engineers would benefit the former greatly.

Despite this, the software for seismic analysis that is available today has not been written by software developers but by seismic engineers who have adapted to – sometimes out-of-date – tools such as Matlab or specific programming languages. It is not usual for seismic engineers to cooperate with, or enlist the help of others in developing applications, possibly because they are not yet aware of what can be achieved or improved.

In this context, it is hoped that SelEQ, described further ahead in Chapter 3, being an innovative application, will not only simplify several processes used in structural analysis but also serve as a proof of concept for what can be achieved through cooperation.

SelEQ aims to be available on the World Wide Web for everyone, serving as a means to promote more interactivity between seismic engineers and software developers, as well as being a useful tool in seismic analysis.

1.3 Objectives

The main objective is to develop an innovative application that will make searching and finding adequate records significantly easier than it is at the moment. The full specification can be found under Chapter 3, including the formal system requirements (Section 3.3, as well as the use cases (Section 3.4) for all the functionalities planned.

As described on the previous section, it is also hoped that this application will lead to other Web applications for seismic engineers. The Web is an almost completely unexplored field in this domain, and the potential for expansion and development of interactive, cooperative tools is immense.

The application developed is designed to be extensible, allowing anyone with a basic knowledge of programming languages to expand upon it easily. This is to encourage development of new functionalities and applications with a similar business model.

1.4 Structure of the Dissertation

In addition to this introduction, the dissertation consists of five chapters.

Chapter 2 describes the problem at hand, along with particular domain concepts, as well as similar applications and the current state of the art on Web applications.

Introduction

On Chapter 3, the full specification of SeIEQ, the application that was developed as proof-of-concept, is exposed. Both seismic and formal requirements are included, as well as the use cases.

Chapter 4 discusses the details regarding the implementation of SeIEQ, namely the architecture, the structure of the database and the programming languages used.

Chapter 5 shows the evaluation of the work and presents a case study.

Finally, the main conclusions of the work are located in Chapter 6, along with recommendations for future work in the field.

References are included at the end of the document.

Introduction

Chapter 2

Web Application for Earthquake Record Selection

While seismic engineers have at their disposal a wide variety of applications or scripts to assist them in their work, and while these allow for several common tasks to be automated, those are still not enough to satisfy all the needs in that domain.

For certain types of structures the designers need to perform advanced non-linear dynamic analysis which requires a proper selection of earthquake records from databases containing thousands of ground-motion signals recorded during past earthquakes.

The tools available can be divided into two categories: seismic analysis, which focuses on predicting to a degree of certainty the response of a structure to an earthquake; and finding these records from the databases.

Tools for seismic analysis are common – software for dynamic analysis is shared between several areas of applied physics. Tools for selection of records aren't, and the few that exist are written using proprietary software.

The following section describes the problem for which a solution is presented, and it is followed by a section describing REXEL (Section 2.2.3), an application distributed freely on the internet that is the closest to a solution available prior to this dissertation.

Also included is a brief review of Web applications and the current state of the art on the field, as well as free database platforms.

2.1 The Problem

The context in which this application is developed is not a trivial one. There are several notions of physics and the seismic analysis domain that must be understood *a priori*

because they are incorporated somehow in the final product. This section will briefly describe those concepts, as well as the calculations that must be performed when a goodness-of-fit is calculated.

2.1.1 Domain Concepts

Figure 2.1 illustrates the different concepts in this domain and how they relate to each other in a UML Domain Object Model. It is particularly relevant to note the distinction between Accelerogram and Records – each Accelerogram contains three distinct Records (based on a three-dimensional space, each of these corresponds to one axis, so there are two horizontal components and one vertical component to each record), and each Record contains three different types of response Spectra – one for acceleration, one for velocity, and one for displacement – all calculated for a specific *damping* value¹.

The acceleration response spectrum is the most commonly used and is usually provided in seismic design codes such as the Eurocode 8 and ASCE 7-05.²

2.1.2 Goodness-of-fit Calculations

For simple and regular structures, the seismic analysis can be performed by looking at a given structure as an equivalent single degree of freedom system characterised by a mass, a stiffness and a level of viscous damping.

There is no need to find an actual seismic record to perform any calculations. The seismic action is accounted for in the design as a lateral force applied to the mass of the system which is derived based on the fundamental period of the single degree of freedom and on the acceleration response spectrum provided in the seismic design code. It is therefore sufficient to use the acceleration response spectrum calculated directly from the regulations, as described on the previous section, as the force to calculate the response of the structure.

However, for critically important structures such as nuclear power plants, bridges, hospitals, etc, more advanced analysis should be performed which typically involves dynamic analysing that requires record selection.

For this, a goodness-of-fit formula has been established [BSB09][BA04][Cas06]. The closest records to the spectrum given on the regulations should be used for calculations, according to several constraints. The goodness-of-fit is calculated by the root-mean-square difference (D_{rms}), as defined by Expression 2.1.

¹Damping is any effect, either deliberately engendered or inherent to a system, that tends to reduce the amplitude of oscillations of an oscillatory system.

²The regulations for how to obtain these spectra are compiled under the appendices at the end of this report, specifically under Appendix A – for the American regulations – and Appendix B – for the European regulations.

Web Application for Earthquake Record Selection

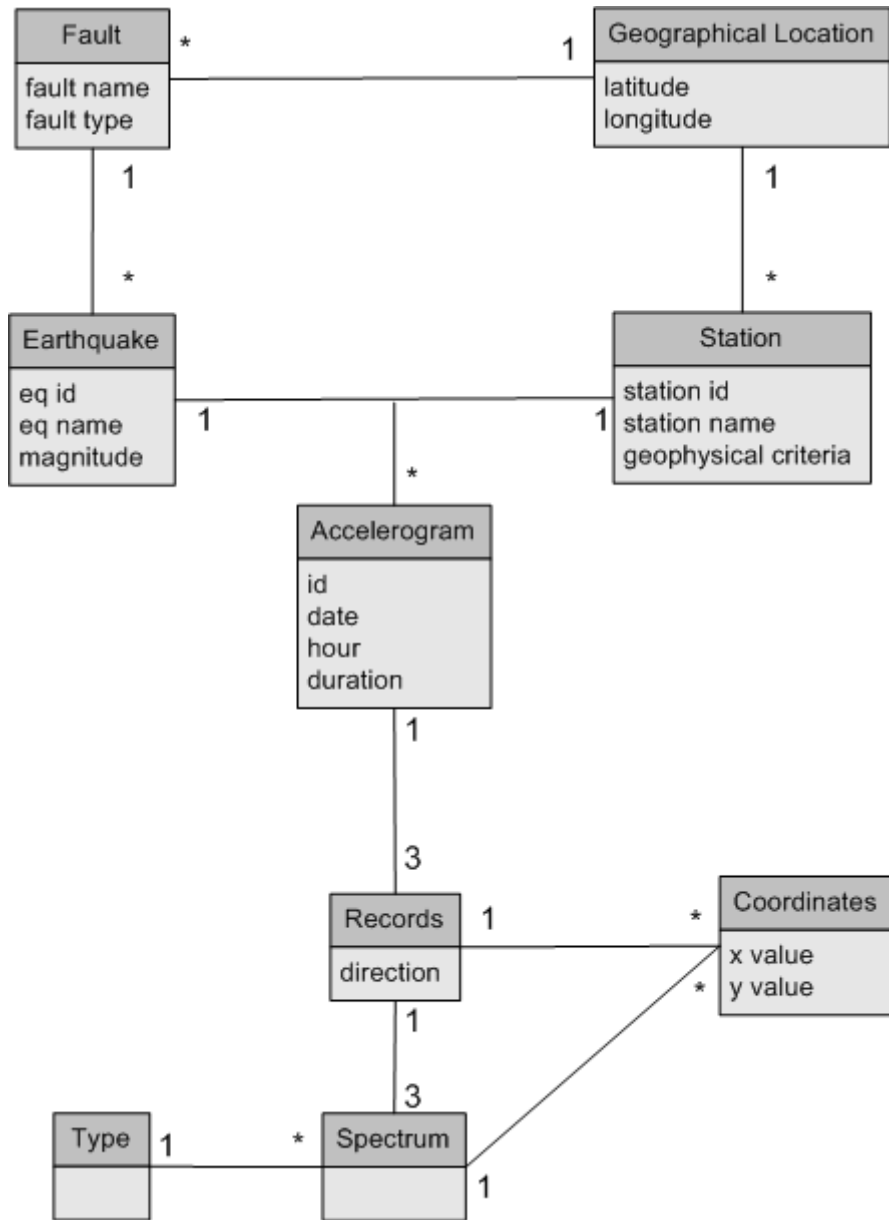


Figure 2.1: Domain Object Model

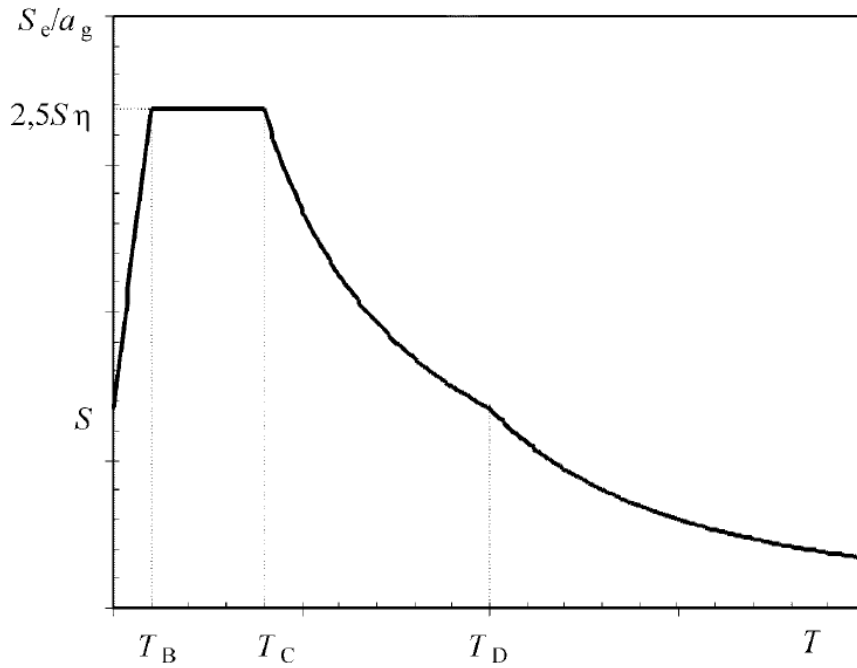


Figure 2.2: Shape of the European Elastic Response Spectrum

$$D_{rms}(\alpha) = \sqrt{\frac{1}{a-b+1} \sum_{i=a}^b (\log_{10}[\alpha \cdot Sa_R(T_i)] - \log_{10}[Sa_T(T_i)])^2} \quad (2.1)$$

The D_{rms} expression quantifies the difference between the target spectrum (the one defined by the regulations) and the spectrum generated for each of the records that comprise a given accelerogram. Referring back to the model in Figure 2.1, each accelerogram contains three independent records (one for each axis). It is the spectra generated from these records that are compared to the target spectrum and evaluated according to their goodness-of-fit.

Often only one of the spectra is the best, as spectra obtained for other directions don't fit the target spectrum as well.

The parameter α is also defined in Expression 2.1, and it is a scaling factor. Because the spectrum considered might have a similar form to the target spectrum, but not the same scale, the α is introduced. This way a spectrum with an adequate form but very low intensity can be considered over a spectrum with similar intensity but an entirely dissimilar form.

2.2 Related Work

This section describes the related work to the Web application suggested. It begins with a short introduction to structural analysis software which explains the context in which this application appears, and describes an application distributed freely on the internet, REXEL, which uses the Matlab environment.

2.2.1 Structural Analysis Software

Seismic engineers have several excellent tools for structural analysis. These tools are shared between the many fields of physics that require dynamic analysis, and have been thoroughly tested as a result.

What is lacking in this domain is input selection. Structural analysis in the seismic engineering domain requires that actual seismic records, harvested from real occurring earthquakes, be used for complex structures. This selection of records is an almost unexplored field.

2.2.2 Matlab

Matlab [Mat09] is a computing environment used for applied mathematics. It allows its users to create scripts to facilitate repetitive tasks, or run a specific algorithm several times. It also allows creation of user interfaces, and interfacing with other languages. For its incredible power, Matlab is widely used in academia.

The Matlab Component Runtime can enable users to create applications in languages such as .NET or Java, and use Matlab functions to compute specific tasks. It is the case with REXEL, described in the next section.

2.2.3 REXEL

REXEL [IGC08] is a software written for Matlab, distributed on the internet. Included with the installation package is a searchable database, containing the European Strong-motion Database and the Italian Accelerometric Archive.

While this application is incredibly powerful and the only one existing in its field, it is still not ideal as a solution to input-searching. There are several problems with the architectural design of REXEL as it is currently distributed.

REXEL requires that the user have installed a compatible version of Matlab (which is proprietary software and therefore not easily available to everyone), or that the Matlab Component Runtime is installed.

Being a desktop application, REXEL does not allow for easy changes or updates to its database or to the business logic. Since REXEL must be downloaded and installed locally, any time changes are made to its logic the users must download the new version,

including all of the database. Similarly, whenever the database is updated, the users must obtain the newest version by downloading and re-installing the full application. REXEL also requires the user to register an account before downloading the application.

2.3 Web Technologies

There are many different technologies for web development and publishing today, each with its own pros and cons. This section aims to give a brief overview of three major development technologies, finishing with a comparative analysis of each along with priorities regarding the various attributes, justifying the choice for this project.

The three most used programming languages for the web were used in this analysis.[LLC09] The MVC Architectural Pattern was also analysed for its prominent use in Web applications to better isolate the business rules from the presentation (Section 2.3.1).

2.3.1 Model-view-controller Architectural Pattern

The MVC Architectural Pattern is used in software engineering to isolate the business logic from the user interface. [Ree03] MVC was first described in 1979. [Ree79]

There are three components to this pattern, and they interact always in the same way. The *model* handles application and business logic, the *view* handles presentation logic, and the *controller* accepts and interprets the input. On a web platform, the controller serves as an entry point, and makes the calls to the model – which analyses the data and performs calculations – and the view – which handles the data returned by the model and generates the user interface – returning the interface to the user.

Figure 2.3 describes this behavior for a web platform.

There are many frameworks³ influenced by the MVC pattern, ranging from Ruby on Rails (written using the Ruby programming language), to PHP-based frameworks.

2.3.2 ASP.NET

ASP.NET [Cor09a] is a framework by Microsoft to allow for web application and web services development. It is released with the .NET Framework [Cor09b].

Applications written for this framework are scripted, although it has transparent integration with C# modules that are typically compiled. It is strongly typed, and also static typed.

The framework itself has a vast number of modules that can be easily added to projects. Many common tasks and features are included in these modules, making it a good development tool.

³A framework is an abstraction in which common code providing generic functionality can be selectively overridden or specialized by user code providing specific functionality.

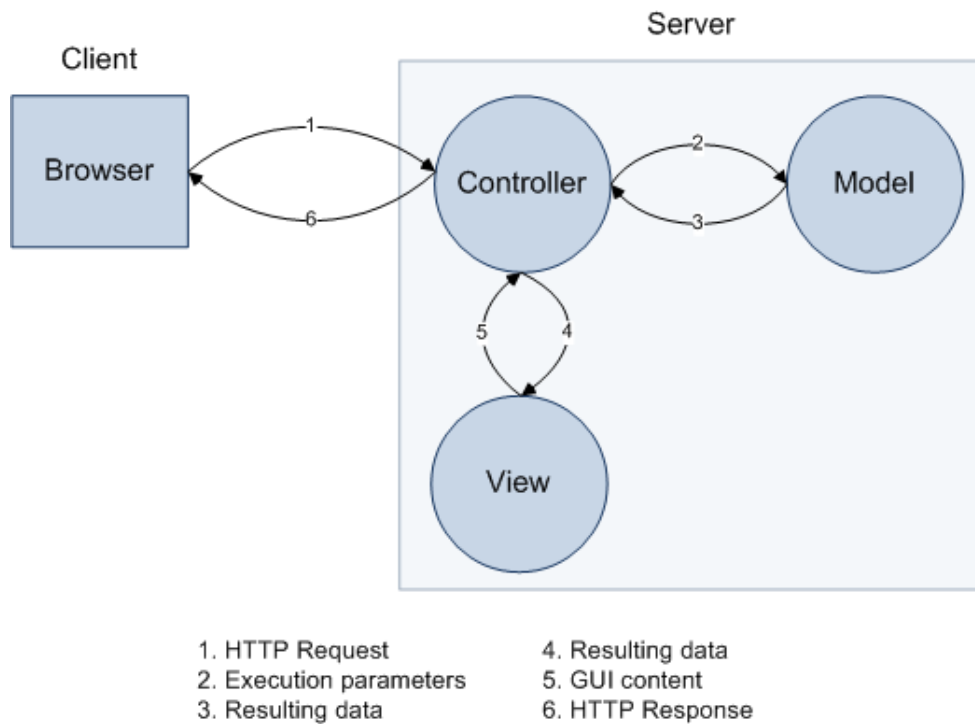


Figure 2.3: Model-view-controller Pattern Behaviour

ASP.NET is proprietary software. It is necessary to acquire commercial licenses in order to install and use an application written for this framework. Furthermore, while it is possible to run the software under Unix operating systems, to use all of its functionalities and modules the server must be running a version of Microsoft Windows, also proprietary software.

2.3.3 PHP

PHP [Gro09a], which stands for PHP: Hypertext Preprocessor, is an open-source scripting language, especially suited for web development. It is a weakly typed language – in the sense that it supports implicit type conversions, and also dynamically typed – given that its type-checking is performed at run-time.

PHP has a vast knowledgebase online, and contributors from all over the world cooperate in building the PHP Manual, leaving comments and snippets of code where relevant to improve the manual’s information and to help the users that browse its pages.

In addition to this, PHP also has an incredible amount of extensions, which are also freely available, that range from database APIs⁴ to graphical libraries, audio manipulation, mail functions, mathematical calculations, and even cryptography.

It is currently the most used language on the internet [LLC09].

⁴Application Programming Interface

2.3.3.1 PHP MVC Frameworks

There are several PHP frameworks that follow the MVC pattern. The most commonly used ones are CodeIgniter⁵, Symfony⁶, Zend⁷ and CakePHP⁸.

Web frameworks can be of great use when developing a dynamic web application, as they can significantly reduce the overhead associated with designing, implementing and testing certain key components like database access and page templating. However, when developing smaller applications, they can actually increase the overhead. In a midsized application, the time spent getting familiar with the framework is usually easily compensated by the built-in functionality, but in smaller and more focused applications this can easily outgrow the resources necessary to develop the base components from scratch.

From a server-side perspective, a Web framework can increase significantly the overall memory footprint of the application as its codebase also needs to be interpreted by the Web server. This affects the server's memory usage and adds some extra latency between requests due to the extra time required to load the application.

2.3.4 Ruby on Rails

Ruby on Rails [Tea09] is an emerging open-source web framework for the Ruby [Sla08] programming language.

Ruby on Rails includes several features such as minimal configuration (valuing convention over configuration) and *scaffolding*⁹.

While it is an incredibly powerful framework – with several built-in functions that make development extremely agile – it is not adequate for an application that should be as extensible as possible for someone with minimal knowledge of programming (part of the Objectives 1.3 is to make the application a proof-of-concept that encourages seismic engineers to migrate their applications to the web).

2.3.5 Comparison and Conclusions

Figure 2.4 illustrates how the different programming languages relate to each other. Each has its own pros and cons - PHP is scalable and familiar, ASP.NET has excellent development tools, and RoR is extremely agile. However, the attributes in this analysis must be prioritised, and it follows from the Objectives in section 1.3 that scalability, maintainability, and popularity are the most important features of the technology to use.

⁵<http://codeigniter.com/>

⁶<http://www.symfony-project.org/>

⁷<http://www.zend.com/en/>

⁸<http://cakephp.org/>

⁹The ability to automatically generate the common create-read-update-delete (CRUD) code the developer can use to build a database-driven application upon.

Web Application for Earthquake Record Selection

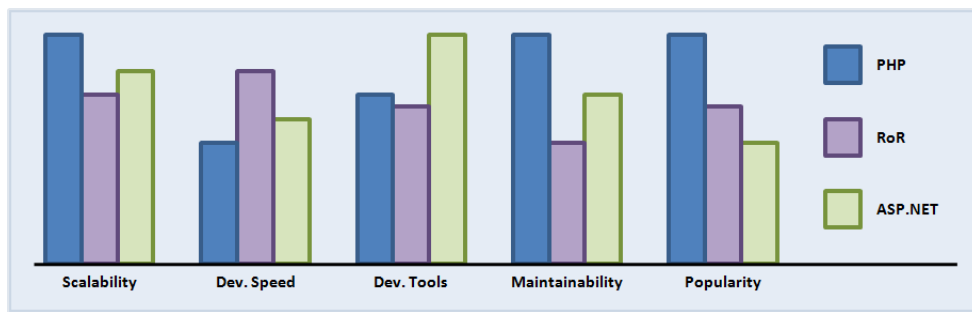


Figure 2.4: Comparison of Programming Languages

Given this, and from the web technologies that were analysed, the best that was identified for this particular application was PHP (section 2.3.3), not only for its open-source nature (ASP.NET (section 2.3.2) is proprietary software and requires commercial licenses, and cannot be used for an application that is expected to run on open-source systems), but also for the vast wealth of information and support available on the internet (something Ruby on Rails (section 2.3.4) is only beginning to develop).

2.4 Relational Database Systems

There are two main relational database systems in use today that are free and open-source, MySQL and PostgreSQL. This section aims to describe the two, as well as offer a useful comparison guide, ending with conclusions about their interest to this application.

2.4.1 MySQL

MySQL [AB08] is a free, open-source relational database system. It is owned by MySQL AB, and its source code is available under the GNU General Public License.

MySQL implementations run in two layers - the upper SQL layer accepts and interprets the queries, and the engine layer runs and stores the information. Because of the vast differences between the engines that MySQL currently implements, it is necessary to specify which engine is being analysed.

The most commonly used storage engines are MyISAM and InnoDB.

2.4.1.1 MyISAM

MyISAM is an engine optimised for reading data as opposed to writing or stability. It does not support ACID transactions [Gra81]¹⁰ nor does it implement foreign keys, and

¹⁰Atomicity, Consistency, Isolation, Durability

it does not support backups or point-in-time recovery natively (however, these can be implemented on the server and often are).

Because it has so few functionalities, MyISAM is one of the fastest serving databases. Without the heavy foreign keys or transactions the processor has more power to comply with the requests.

2.4.1.2 InnoDB

InnoDB supports ACID transactions at the cost of performance and speed when compared to the MyISAM engine. It is a secure, stable engine, optimised for large volumes of data.

2.4.2 PostgreSQL

In contrast with MySQL, PostgreSQL is a database server with a single engine. [Gro09d] It implements most state of the art functionalities available to databases, including backups, point-in-time recovery, foreign keys, all sorts of indexes, and several storage methods.

PostgreSQL isn't developed by any one company – it has several developers and companies actively engaged in the development of new releases. As a result, there is vast support online from this community.

2.4.3 Other Database Systems

Other database systems were analysed, but dropped early in the development because of their costs. Oracle Corp.'s Oracle, Microsoft's SQL Server, and IBM's DB2 are all proprietary software. SQL Server in particular will only run under Microsoft Windows.

2.4.4 Comparison and Conclusions

Because in such an application speed is of the essence, and the database does not need support for heavy queries or foreign keys to ensure stability of data, it is recommended that MySQL, with the MyISAM storage engine, be used.

While most commonly used functionalities are implemented by PostgreSQL or even MySQL's InnoDB engine, these sacrifice a part of processing speed. Since the data saved on the database is not to be interacted with, speed of access is considered to be of greater importance.

2.5 Conclusions

While there is a vast range of software aimed to seismic engineers, from scripts to calculate specific formulas to full applications, these are still not ideal.

Web Application for Earthquake Record Selection

The business logic of an application should be easy to change or improve (the application should be *extensible*), and the dataset should be centralised, not distributed with the application, so it can be more easily updated, with the changes taking effect for all users with minimum or no hassle.

PHP proves to be ideal for the development of an application in this domain, for its ease of use, the similar syntax to other programming languages seismic engineers might be familiar with, and the large knowledgebase and support on the internet.

MySQL, with the MyISAM engine (see section 2.4.4), has all the features the database would need (read and write of data), allied to an unparalleled speed. While PostgreSQL and the InnoDB engine for MySQL both implement several functionalities not available in MyISAM, these are not of the essence.

Web Application for Earthquake Record Selection

Chapter 3

SeleEQ Specification

SeleEQ is the name of the proposed application. It stands for *Select Earthquake*, in a direct reference to the ability of the application to select seismic records from a large database.

This chapter describes the full specification of the application, including the formal requirements and use cases. Specific calculations, such as definitions for target spectra in the American and European models, are included in the appendices.

3.1 Introduction

The silver bullet to the problem described throughout this dissertation has some very particular characteristics. First and foremost, it must not force the user to have any specific hardware, or require that the user install any supplementary software in order to use it. It must use software that is commonly installed on most computers (in this case, a Web browser).

Any updates to the database must be transparent to the users, that is, users must not make any conscient effort to use the newest available version of the records or even take note of changes. Similarly, whenever the logic behind the application is changed, the users should not be affected by this and should continue to use the application uninterrupted.

It is also desired that the application runs on free, open-source software that anyone can install.

3.2 Seismic Engineering Requirements

There are a few requirements that are major for the application, and they come from a seismic engineering domain. These are explained in further detail on the following subsections, but a short explanation is given here.

Although the domain of structural analysis is incredibly rich in multiphysics applications¹, seismic engineers are still lacking in tools to enable them to choose the input to the applications for analysis (Section 2.1).

Seismic records are typically saved as accelerograms, with three separate records (corresponding to the three dimensional axis). These records can be used individually depending on their goodness-of-fit. One of the horizontal components might be better suited for a particular calculation than the other, so this must be taken into account.

For the goodness-of-fit calculations, a target spectrum is required. This target spectrum depends on the country's regulations, and the application must include at least the American regulations [oCE03], as by the American Society of Civil Engineers, and the European regulations, stated on Eurocode 8 [fS04]. In addition, and to allow for maximum flexibility, the application must also include the possibility of inputting a user-defined spectrum, for compatibility with all other cases².

All records must then be evaluated and a corresponding spectrum calculated from them. This spectrum is calculated according to a specific formula, which takes into account the damping value selected. Regulations establish this damping value depending on several factors, and it is one of the input values the user supplies.

The application should, therefore, be able to calculate the spectrum that belongs to a particular record, and then compare it to the target spectrum, using the goodness-of-fit formula.

3.2.1 Filtering of Data

The application should allow the user to filter, ahead of time, records that may be undesirable according to geophysical criteria. For example, a user may have no interest in records obtained more than 50km from the hypocenter, or desire only records that have been obtained for a particular fault.

The user should be able to choose what criteria to filter for.

3.2.2 Calculation of Spectra

The application should also calculate, in a manner that is transparent to the user, spectra that relate to the user's specified damping values for each of the filtered records. These calculations should be performed when they haven't been before – the application should keep a cache of records and their corresponding spectra for damping values that have been previously selected.

¹Multiphysics applications – applications used in several fields of physics, such as civil or seismic engineering.

²In case a user chooses to adapt the results to a spectrum that cannot be defined by American or European regulations, such as Japanese or Italian regulations, the application must allow the user to specify an array of points that define it.

3.2.3 Goodness-of-fit Calculations

The application should allow for D_{rms} input selection, as well as calculations of D_{rms} values for selected records. It should also be possible to display records by their goodness-of-fit, so the fittest may be selected for exporting.

3.2.4 Exporting Records

Records that are deemed fit, or fittest, from the selection the user arrives to, must be retrieved from the application. The user should be able to select what information is important to retrieve, and select to export records. The application should return these records in the appropriate format according to the data to export that has been selected.

3.3 System Requirements

This section describes formally and in detail the system requirements of SeIEQ, including functional and non-functional requirements alike.

3.3.1 Non-functional Requirements

The application has several non-functional requirements, that have already been described previously to varying levels of detail on chapter chap:sota. This section aims to describe them briefly.

3.3.1.1 Extensibility

The application is *extensible*. It is easy to add another module without having to alter the pre-existing code. This new module must be able to access the same data and use the same interface as the rest of the application, being fully integrated into its code.

3.3.1.2 Maintainability

The application is *maintainable*. It is not be exceedingly complicated to change the logic of the application, and it is not be necessary to change more than the business logic files for this. Similarly, when changes are made to the GUI³, it is not necessary to change the business rules.

The application must be developed in a well-known and commonly used programming language, with a vast wealth of information available, so as to maximise its maintainability.

³Graphical User Interface

3.3.1.3 Speed Considerations

Efforts are made to guarantee that the application does not take more than the reasonable time to respond to the users' queries. Because several operations require many calculations, these are optimised as possible, and the server should respond in a timely manner.

3.3.2 Functional Requirements

3.3.2.1 Filtering of Records

Add Filter The SelEq application supports the addition of search filters (such as magnitude or distance to the fault) in order to filter the records retrieved from the database. When selected, this filter is added to a list on the main page, allowing the user to add his own data for each filter.

Remove Filter The SelEq application allows the user to remove any previously added filter. If the user wishes to use the filter at a later date it will have to be re-added.

Use D_{rms} SelEq is prepared to allow the user to order records by their D_{rms} ⁴, an important measurement for seismic engineers. This function requires additional input from the user relative to the target spectrum. If use of D_{rms} is selected on the main page, the application prompts the user to insert this additional input.

Deactivate D_{rms} If D_{rms} is active, the user can opt to remove it by pressing a cancel button. If D_{rms} information is removed from the search, it will be impossible to order retrieved records by their D_{rms} until it is made active again.

3.3.2.2 Ordering of Results

Order by Column The search results page allows the user to order records in an intuitive manner – by selecting the column header, where the column name is placed. It is possible to order records by magnitude or distance, as well as any other geophysical characteristics associated to the records. If D_{rms} is in use, then the D_{rms} column is initially defined as *infinite* for all records.

Calculate Spectra When D_{rms} is in use, the application calculates the spectra to be compared to the target transparently, from the seismic records. For efficiency purposes, these spectra are saved when they are calculated so they don't need to be re-calculated at a later

⁴ D_{rms} – root mean square difference

date for the same values. These spectra calculated only when necessary – the application should evaluate them lazily⁵.

Calculate D_{rms} If the D_{rms} function is in use, each row of results retrieved contains a checkbox for calculating the Drms. The user can select up to five records and request their Drms to be calculated versus the target spectrum previously defined.

Change D_{rms} Information It is possible to alter the previously established spectrum and the target interval to be used on the search records page. If the user chooses to do this, the Drms column will be re-initialized to *infinite* for all available records.

3.3.2.3 Exporting Selected Records

Exporting Records Each row of the retrieved records table contains a checkbox for exporting data. The end user can select as many records as desired from the retrieved records and the information requested from them (check domain object model). The application then returns a .zip file containing one .zip file per record, with the corresponding .csv files within (one per graph).

View Record Each row contains a button for viewing the corresponding record, which displays a window with several information regarding the individual record. This window also displays a button for export of a singular record.

Export Singular Record When the user selects to view the record, he has the possibility to view a runtime generated image or download individual .csv files containing the record data. The user is prompted as to what information he wants to download relative to that particular record, similar to the multi-record export feature.

3.4 Scenarios of Usage

Given that there is only one actor in the system, the use cases are very similar to the functional system requirements. They are included here for completion.

Figures 3.1, 3.2, and 3.3 describe, the same way they were discussed before, the Filtering, Ordering, and Exporting of records, from a user's point of view. They show the entire list of use cases for the SeIEQ application.

⁵Lazy evaluation refers to the property of a language or system to not evaluate a computation until it is needed. In this case, it refers to the fact that a spectrum won't be calculated from the original harvested files until there is a user that requests them, at which point it will be created and stored in memory for posterior use. This keeps the necessary memory for the data to a minimum, as only spectra that have been requested will have been created and stored.

SeIEQ Specification

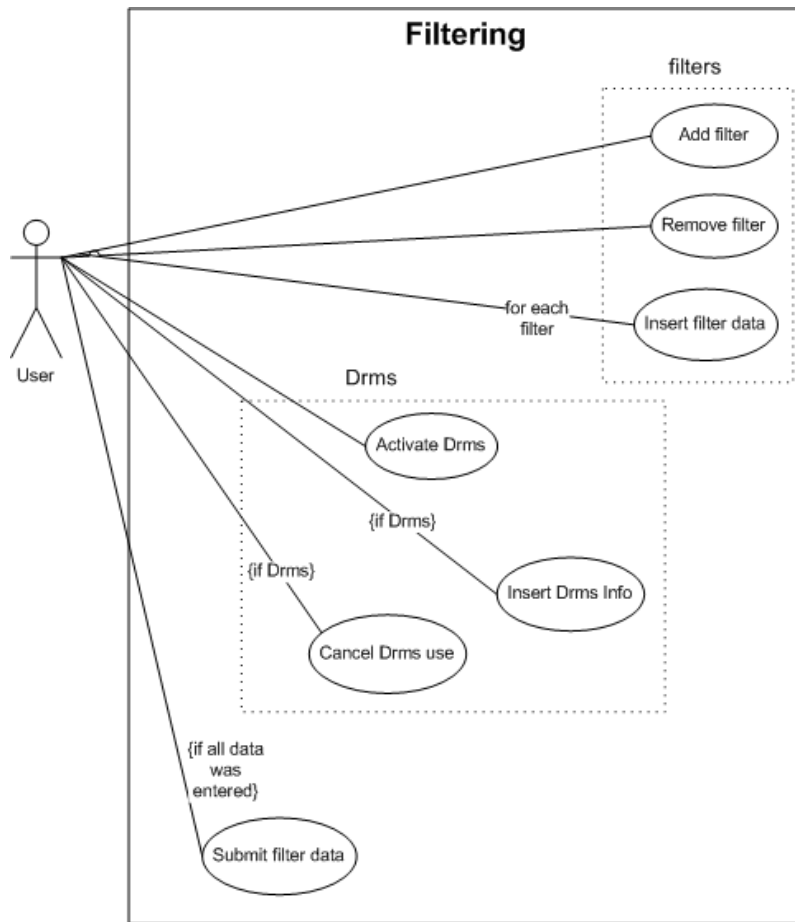


Figure 3.1: Use Case 1 – Filtering

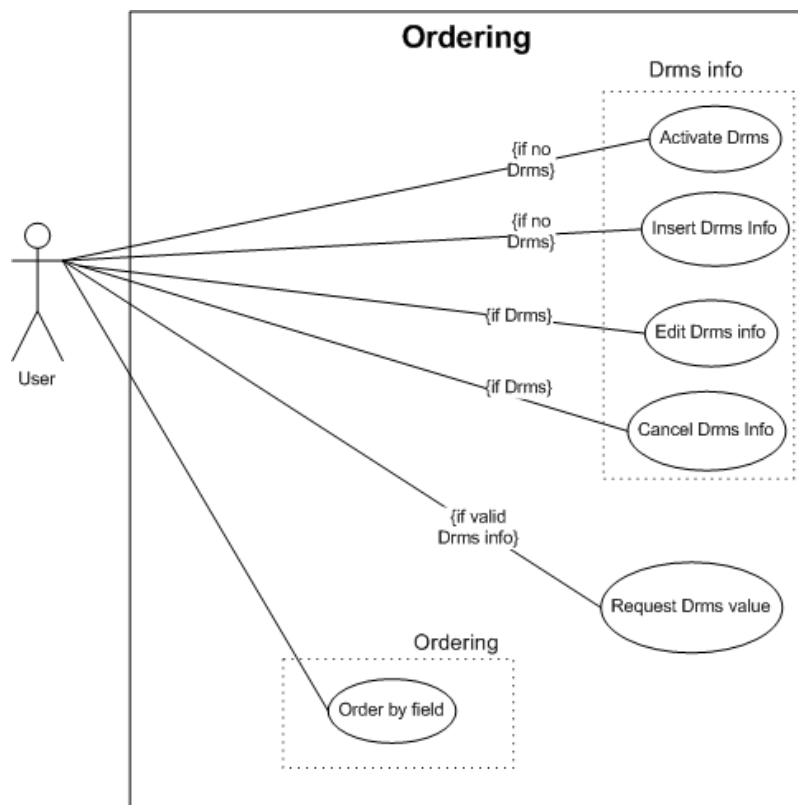


Figure 3.2: Use Case 2 – Ordering

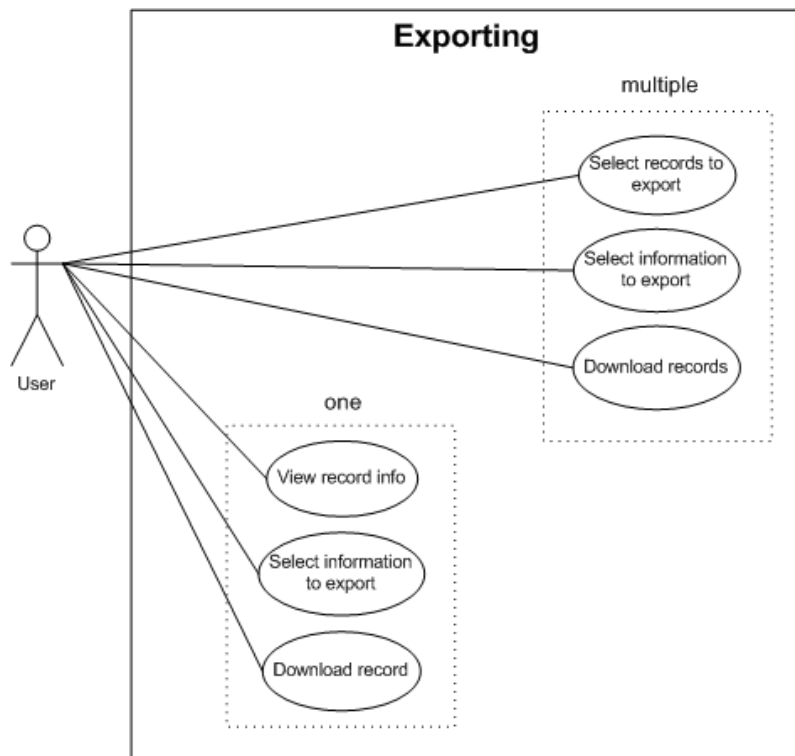


Figure 3.3: Use Case 3 – Exporting

Not included is the insertion of records into the database, since it happens externally to the application and is not part of the specification, as well as the parsing of files, which is part of the initial set-up.

3.5 Conclusions

The main obstacle in understanding the problem and formulating the requirements is clearly the domain, which is rich in concepts and terms that aren't very intuitive for a developer. Another big obstacle is understanding the American and European regulations regarding seismic analysis without being a seismic engineer.

The formal requirements naturally follow from the seismic requirements, which are in turn a formulation of part of the needs seismic engineers have in terms of finding the ideal records for their calculations. But apart from simply searching for, and selecting, records, the SeIEQ application also serves as a strong foundation from which to build larger database-driven web applications.

Chapter 4

Design & Implementation

This chapter focuses on the design and implementation details of the application described on Chapter 3. It includes the architecture design, information regarding the application data and the data model, the programming languages and applications used, and finalises with conclusions regarding the implementation of SeIEQ.

4.1 Architecture Design

SeIEQ is based on a Web client-server, 3-tier architecture (Figure 4.1), where a client (the user) makes requests to a Web server that calculates and returns data. This data can be returned in the form of Web pages (for the user to continue to interact with the application) or files (containing the information selected by the user for export).

The Web server is typically a computer connected to the internet with a Web server application such as Apache (Section 4.1.1) installed and running, along with a PHP module for interpreting the SeIEQ scripts, and with a physical connection to a database server such as MySQL, which contains the actual searchable data.

4.1.1 Web Server – Apache

Apache [Fou09] is a free, open-source server that can be installed on both UNIX and Windows systems. It is a secure, efficient, and extensible server. Apache is the most widely used Web server today.

In this context, Apache is installed with a PHP module, allowing it to interpret in runtime the requested PHP scripts, whose output is returned to the user in the form of a Web page the user can then interact with.

Apache also handles all calls to static files, such as the CSS (Section 4.3.2.2), image and JavaScript (Section 4.3.1.2) files, returning them to the users as necessary.

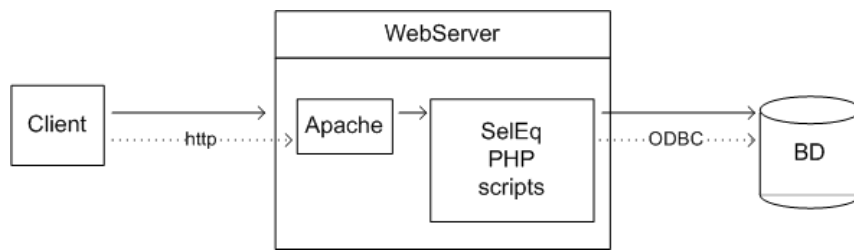


Figure 4.1: SelEQ Architecture

4.1.2 Protocols

A protocol is a set of rules enabling two computers, applications, or processes to communicate with each other. A protocol defines how the connection, communication, and transfer of data must be processed.

SelEQ makes use of two different protocols. Their use to the application is described in the following subsections.

Figure 4.1 illustrates the architecture model, clearly displaying the protocols used.

4.1.2.1 HTTP

SelEQ uses the HyperText Transfer Protocol (HTTP) to transfer both the seismic data and the interactive webpages from the server to the user, according to the commands the user issues. This protocol is used for all communication between the user's machine (the *Web browser*) and the server's application (the Apache Web Server – Section 4.1.1).

4.1.2.2 ODBC

The ODBC protocol is used by the PHP scripts to communicate with the MySQL database. PHP implements full ODBC support [Gro09c], and the MySQL server was installed with ODBC support.

4.2 Application Data & Data Model

The seismic records have most of their information stored in a MySQL database, including all geophysical criteria such as magnitude, latitude and longitude, fault distance, etc. These are the records the application allows the user to filter for. The remainder of the data, corresponding to the actual seismic records, is stored in binary files outside of the database.

SeIEq includes an easy to use parser to transform .csv database files into sql queries, which can optionally be added directly to the database if the connection information is supplied. The file is optimised for the NEHRP database¹ [oC09].

For each installation of the application, a database handler file must be edited. This file describes the database information that is searchable and retrievable, allowing for maximum flexibility in the databases used.

4.2.1 Denormalisation

The principles behind normalisation are, according to E. F. Codd: [Cod71]

1. To free the collection of relations from undesirable insertion, update and deletion dependencies;
2. To reduce the need for restructuring the collection of relations as new types of data are introduced, and thus increase the life span of application programs;
3. To make the relational model more informative to users;
4. To make the collection of relations neutral to the query statistics, where these statistics are liable to change as time goes by.

All these principles, while very important in other applications, are not a priority for SeIEQ. First, the data is never updated or deleted from the database, so there are no dependencies; and the information that is redundant (see Figure 4.2) is not sensitive, so even with minor inconsistencies resulting from insertion of data (such as a misspelled earthquake name), the database is still valid and useful to the users.

Secondly, all the searchable information is included on the database. The addition of new domain concepts will not render the current model invalid. While at this time there is no intention to extend the data model, doing so will not affect the application's behaviour.

Similarly to a data warehouse, the intuitive, high-performance retrieval of data supports the decision of denormalisation. In terms of classes, the database is small enough that very little information regarding semantics is lost in this denormalisation. Figure 4.2 and Figure 4.3 show the database before and after normalisation.

An important note here is that while the databases typically model the domain object model (Figure 2.1), several concepts from the DOM are not necessary for the application.

¹National Earthquake Hazards Reduction Program

Design & Implementation

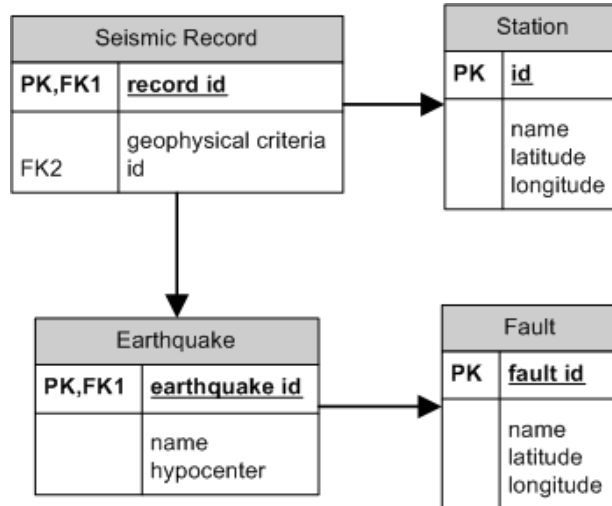


Figure 4.2: Normalised Database

Seismic Record	
PK	<u>record id</u>
	geophysical criteria earthquake name hypocenter fault name fault latitude fault longitude station name station latitude station longitude

Figure 4.3: Denormalised Database

4.3 Programming Languages and Applications

This section describes all technologies used by SeIEQ, from programming to markup languages, as well as the applications developed concurrently to aid in the development and maintenance of SeIEQ.

4.3.1 Programming Languages

4.3.1.1 PHP

SeIEQ is implemented primarily in PHP (described earlier in Section 2.3.3), an interpreted programming language. PHP has numerous extensions, in particular handlers for several databases, as well as libraries of functions for graphics creation and editing.

SeIEQ uses PHP's ODBC extension [Gro09c], as described under Section 4.1.2, to connect to the MySQL database. This extension is used to connect, communicate, and transfer information, in this case, perform queries and receive the results.

In order to generate actual image graphics of the seismic records for viewing, in runtime, the application uses PHP's graphics library GD [Gro09b].

PHP is installed on the webserver as an Apache module (see Figure 4.1 or Section 4.1.1 for more information).

While the MVC model was followed in the development of SeIEQ, no framework was used because of the overhead in performance it would bring (see Section 2.3.3.1).

4.3.1.2 JavaScript

All the client-side interaction is coded in JavaScript. JavaScript is a weakly-typed language, a dialect of the ECMAScript standard, and a commonly used scripting language on the client-side of web applications [McD03].

A free open-source library is used for Document Object Model manipulation and asynchronous communication – JQuery [Res09]. JQuery is used, among other things, to create dynamic forms [CS07], allowing the user to interact with the application without having to refresh the whole page.

AJAX (Asynchronous JavaScript And XML), a group of techniques used in the client-side to add interactivity to web applications without the need to interfere with the existing page, is also used. Requests are sent in the background using the XMLHttpRequest object, and only parts of the page are changed as opposed to a whole new page being requested without the use of these technologies.

AJAX commonly requests XML or HTML data, but JSON², plain text or any other format can be retrieved using this method.

²JavaScript Object Notation

4.3.2 Markup Languages and Style

4.3.2.1 XHTML

XHTML³ [W3C02] is a markup language, and a dialect of XML⁴. It is based on blocks that can contain other blocks, describing a document as a collection of data arranged in a tree structure.⁵

Meta information (like keywords or a description of the content) can be included under the appropriate blocks. Headers and titles are contained in their own blocks, and even paragraphs are described accordingly.

Common elements of a GUI, like buttons, are included in the XHTML specification and are implemented by all major browsers. Included also are table and image elements, as well as paragraphs, headers, and forms. All major browsers implement forms and their objects as interactive controls, allowing the user to insert relevant information and posting back the data to the server transparently, with minimal XHTML syntax.

The version of XHTML used is 1.0 Strict.

4.3.2.2 CSS

CSS (*Cascading Style Sheets*) [W3C09] is a style sheet language that allows its users to add style to structured documents. Its main purpose is to separate presentation style (the style sheet language, which includes the background color, the element margins, the font size, among many other properties) from the content of documents (which is described in a markup language such as XHTML).

SeIEQ makes use of CSS to style the dynamically generated pages. To change the look and feel of the entire application a developer would need only to exchange the CSS file with their own version.

4.3.3 Support Applications

A small PHP script was developed to parse the NEHRP database (a CSV⁶ file) into SQL code, capable of being imported to the MySQL database.

Any user may edit the parser to allow for other CSV files to be easily parsed and added to a database. This involves selecting the relevant fields from the CSV file that should be included on the database, giving the parser appropriate information regarding the database connection, and simply running it in a similar environment to the one described on Section 4.1, where the PHP is allowed to connect to the MySQL host.

³eXtensible HyperText Markup Language

⁴eXtensible Markup Language

⁵XHTML is used over any of the HTML specifications because unlike the HTML specifications, XHTML is valid XML, and can be validated by any XML parser, as well as edited by any of the many XML editors.

⁶Comma-separated values, a file type commonly used for the digital storage of data because it adds very little overhead (a simple comma separates values) compared to other filetypes.

4.4 Calculations

An effort has been made to optimise the calculations as much as possible. A more detailed study of how to achieve this is needed for further optimisation, however, the preliminary results are satisfactory.

The algorithms that calculate the target spectra and the D_{rms} difference are, for the most part, the definitions of the functions themselves. But whenever a number needs to be calculated more than once, because it becomes part of a function, it is saved and its result used instead.

A good example comes from the American regulations, included with this dissertation on Appendix A. The regulations specify that, for example, S_{MS} is calculated from $F_a S_S$ (view Equation A.5). This value is calculated only once for each D_{rms} target spectrum.

Also in the case of the European model, to give another example, the spectra values are extracted from the corresponding table only once and then used in calculations.

When calculating any D_{rms} value, the first step is to retrieve the information from the file containing the spectrum. This information contains several points from a spectrum, in the form of an acceleration versus a period. The period intervals are irregular.

The application then builds a list of accelerations, as defined by the target spectrum, for each of the existing periods on the spectrum file, that is, all the possible values of T are mapped for the target spectrum, so they may be compared. Depending on the regulations, a different mapping function is used.

Finally, the list of values for the target spectrum and the current spectrum are run through the previously defined D_{rms} formula – Equation 2.1.

4.5 Conclusions

Architecture The server-side architecture allows for maximum flexibility in updates. Keeping most of the application's functionalities server-side means any updates to the application or database can be instantly accessible to the users, without them having to change anything on their computers. This is a big improvement over existing scripts or applications, which are typically saved on the users' computers and need to be updated manually.

MVC Pattern The MVC pattern, described in full on Section 2.3.1, means any changes to how the application *looks* (view) are easy to separate from changes to the application's *data* (model). In terms of maintainability and extensibility, this is a very big plus, especially when it is desired that the application be as extensible and maintainable as possible.

Design & Implementation

Application Data and Data Model In terms of data, because the entire searchable database is kept on the server, it can be easily maintained. Whenever new records are available they can be added to the database and be made available to the users immediately.

Chapter 5

Evaluation and Case Study

This chapter makes a comparison between the actual application and the application described under Chapter 3. It also includes a brief evaluation of the application with regards to its extensibility and maintainability, as well. The chapter also includes a case study under Section 5.2.

5.1 Evaluation

5.1.1 Functional and Non-functional System Requirements

All functional system requirements were implemented, with the exception of the calculation of spectra from the data harvested by stations (the records that comprise the accelerograms, see Figure 2.1). SeleEQ is capable of filtering data, calculating D_{rms} information, ordering the results, and exporting several different types of information.

The spectra used were generated *a priori* from a small selection of records. For lack of data, the D_{rms} calculations are therefore based only on these records.

5.1.2 Practical Use

The application developed, at its current development stage, is of limited practical use in the field of earthquake selection, serving more as a proof-of-concept than a full standalone application. This is because the calculation of the spectra from the accelerograms was not implemented for lack of time and because it was identified as a requirement too far into the development.

However, since this data can be calculated externally and added to the application at a later date, the application is still of use. Because SeleEQ is easy to maintain and extensible, it also serves as a good starting point for other Web applications in this domain.

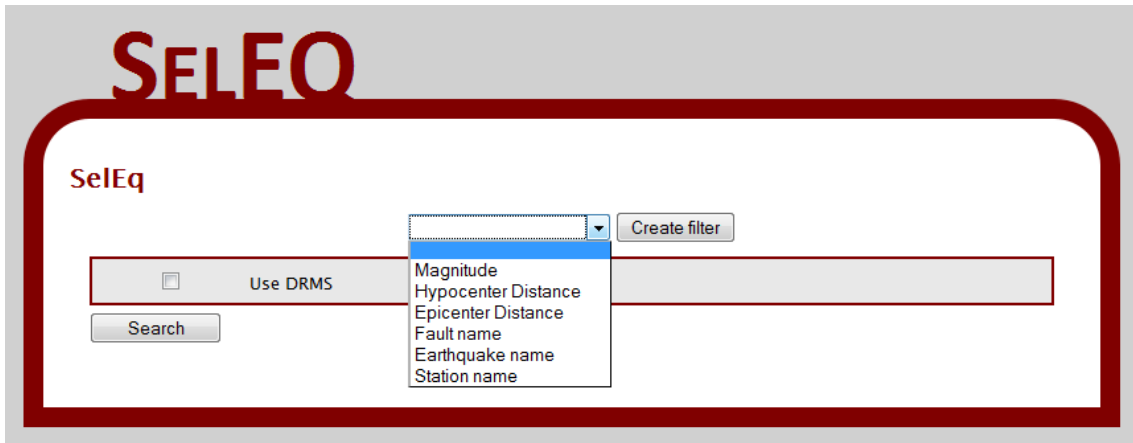


Figure 5.1: Selection of Filters

5.2 Exploitation

As part of the evaluation of the application developed, a sample case study is presented. In this section, a sample scenario is used to illustrate several steps of the application's behavior.

5.2.1 Filtering

The application allows for filtering of records. The user may choose the filters to apply to the results from a list of pre-selected fields. These fields can be changed at any time. Figure 5.1 and Figure 5.2 show this selection and insertion of filter information.

5.2.2 Insertion of D_{rms} Information

The user can then opt to enter D_{rms} information so the application allows for goodness-of-fit calculations (Section 2.1.2). Figure 5.3 illustrates this behaviour for an American spectrum.

5.2.3 Viewing Results

When the user submits the filter and D_{rms} information, a page with the results from the database that comply with the inserted criteria is displayed. Figure 5.4 shows the results, while Figure 5.5 displays the information related to a singular record.

The screenshot shows the SELEO web interface. At the top left is the logo "SELEO" in large red letters. Below it, the text "SelEq" is displayed. A "Magnitude" dropdown menu is set to "Magnitude" with a "Create filter" button to its right. Below this are two filter rows: "Hypo distance (km)" with a range "< x < 4.5" and a "Remove filter" button; and "Magnitude" with the value "6" and a "Remove filter" button. A checkbox labeled "Use DRMS" is currently unchecked. At the bottom left is a "Search" button.

Figure 5.2: Insertion of Filter Information

The screenshot shows the SELEO web interface with the "Drms calculations" section expanded. The "Hypo distance (km)" and "Magnitude" filter rows are visible at the top. The "Drms calculations" section contains a "Type:" dropdown menu set to "American standard". Below it are several input fields: "Site class" set to "A", "Ss" set to "0.2", "S1" set to "0.3", "TL" set to "4", and "Damping" set to "2%". A "cancel drms" link is located at the bottom right of this section. A "Search" button is at the bottom left.

Figure 5.3: Insertion of D_{rms} Information

SELEO

Filters

ID	EQ name	Station	Year	M/D	Magnitude	Drms
<input type="checkbox"/> <u>7</u>	Northwest Calif-02	Ferndale City Hall	1941	209	6.60	<u>∞</u>
<input type="checkbox"/> <u>8</u>	Northern Calif-01	Ferndale City Hall	1941	1003	6.40	<u>∞</u>
<input type="checkbox"/> <u>9</u>	Borrego	El Centro Array #9	1942	1021	6.50	<u>∞</u>
<input type="checkbox"/> <u>20</u>	Northern Calif-03	Ferndale City Hall	1954	1221	6.50	<u>∞</u>

4 records found.

Export Selected

[Return to filters](#)

Figure 5.4: Viewing of Results

SELEO

Filters

ID	EQ name	Station	Year	M/D	Magnitude	Drms
<input type="checkbox"/> 7	Northwest Calif-02	Ferndale City Hall	1941	209	6.60	<u>∞</u>
<input type="checkbox"/>						<u>∞</u>
<input type="checkbox"/>						<u>∞</u>
<input type="checkbox"/>						<u>∞</u>

4 records

Export S

Record #7

Hypocenter Latitude: 40.70

Hypocenter Longitude: -125.40

Fault name:

Time: 1941/209 945

Epicenter Distance: 97.00

Hypocenter Distance: 0.00

Station Name: Ferndale City Hall

[close](#)

Figure 5.5: Viewing an Individual Record

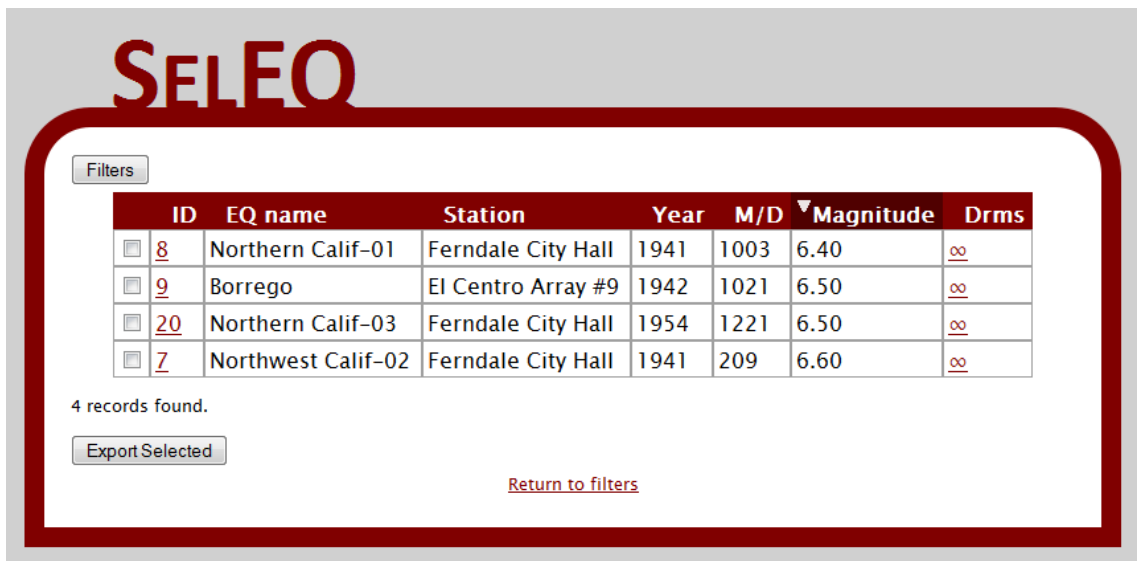


Figure 5.6: Results Ordered by Magnitude

5.2.4 Ordering Results

Clicking on each of the several column headers will order the results by their corresponding attributes, in descending order. Selecting the same column again will order results in ascending order. Figure 5.6 shows the same result set ordered by Magnitude.

5.2.5 Calculation of D_{rms}

If D_{rms} information is included, the application will display an extra column, the D_{rms} column. The D_{rms} data for all the results is initially set at infinity (∞). Clicking the ∞ symbol displays the D_{rms} value for that particular record, as well as the α value that minimises it. Figure 5.7 shows the first record with its D_{rms} information.

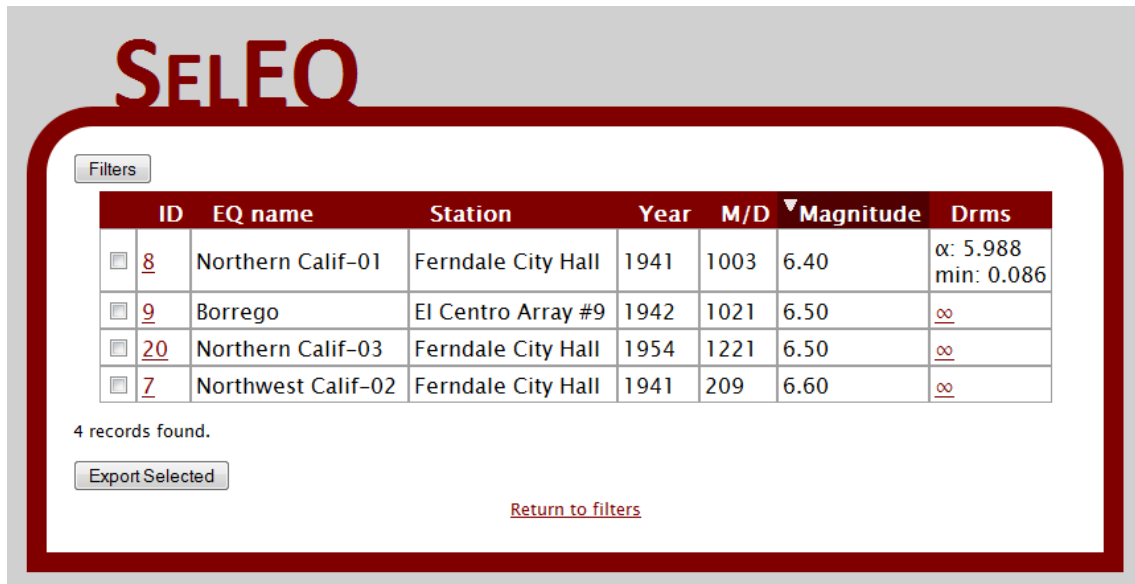


Figure 5.7: D_{rms} calculation

Chapter 6

Conclusions and Future Work

SeleEQ was developed according to the specification in this dissertation, and with the exception of spectra generation (see section 3.2.2), all features were implemented. The final application is capable of filtering according to geophysical criteria, allowing for selection and ordering of the retrieved records, and allow the user to perform goodness-of-fit calculations according to the D_{rms} input given, as well as exporting the best records.

In addition, the application was implemented with common and popular technologies, with care given to its expansibility and maintainability, so it could be expanded at a later date.

A full list of features is described on Formal Requirements 3.3, and the details of the application's design are under Chapter 4.

6.1 Achievements

For the most part, the proposed deadlines were kept in the development of SeleEQ. All requirements were implemented with the exception of spectra calculation. Some spectra were generated from the respective input files (accelerograms obtained from stations during earthquakes) and added to the application to illustrate the D_{rms} calculations.

The resulting application and this dissertation aim to provide a solid base from which it will be possible to develop other applications.

6.2 Future Work

There is a lot of room for expansion in the seismic engineering domain, especially in what concerns selection of input for dynamic structural analysis. While SeleEQ is a step towards

the Web, there is certainly a lot of potential that could come from the implementation of more features.

Since SelEQ is web-based and has been developed to allow for easy extensibility, it is hoped that at some point most of the functionalities listed here will be implemented. These are, however, just a limited selection.

6.2.1 Generation of Spectra

An original requirement that was not implemented, automatic generation of spectra from the harvested records will allow for different values of *damping* to be used, and for new records to be calculated whenever they are necessary.

6.2.2 Filtering by Ground Type/Site Class

Filtering is based on the type of the attribute to filter. If the type is numeric, then filtering is used with numeric values. If the attribute is a name, such as a fault name, then string comparison is used. It would be useful to allow the user to filter by Ground Type. This would require a way to specify logical combinations, since the American and European classifications of ground are different (American Site Class A does not correspond to European Ground Type A), so the user could specify Site Class A *or* Ground Type B, for example.

6.2.3 Geospatial Search

Most of the harvested accelerograms contain the geographical coordinates of the station where they were recorded. Mapping of these on Google Maps¹ or allowing the user to search for records close to a particular location could significantly improve searches for users in some cases.

In this case, the use of PostgreSQL instead of MySQL should be considered, because of the PostGIS support for geographic objects and related queries[DB09].

6.2.4 Generate PDF Report

A functionality to generate a downloadable PDF report of the accelerogram, including all available information, could be added easily.

¹<http://maps.google.com/>

References

- [AB08] MySQL AB. Mysql 5.1 reference manual, 2008. <http://dev.mysql.com/doc/refman/5.1/en/index.html>, accessed on June 2009.
- [BA04] Julian J. Bommer and Ana B. Acevedo. The use of real earthquake accelerograms as input to dynamic analysis. *Journal of Earthquake Engineering*, 2004.
- [BSB09] Nicola Buratti, Peter J. Stafford, and Julian J. Bommer. Earthquake accelerogram selection and scaling procedures for estimating the distribution of drift response. *Under review at the Journal of Structural Engineering*, 2009.
- [Cas06] José Miguel Castro. *Seismic behaviour of composite moment-resisting frames*. PhD thesis, University of London, 2006.
- [Cod71] Edgar Frank Codd. Further normalization of the data base relational model. *Courant Computer Science Symposia 6, Data Base Systems*, May 1971.
- [Cor09a] Microsoft Corp. Microsoft asp.net, 2009. <http://www.asp.net/>, accessed June 2009.
- [Cor09b] Microsoft Corp. Microsoft .NET framework, 2009. [http://msdn.microsoft.com/pt-pt/netframework/default\(en-us\).aspx](http://msdn.microsoft.com/pt-pt/netframework/default(en-us).aspx), accessed June 2009.
- [CS07] Jonathan Chaffer and Karl Swedberg. *jQuery Reference Guide: A Comprehensive Exploration of the Popular JavaScript Library*. Packt Publishing, 2007.
- [DB09] Paul Ramsey Jeff Lounsbury Chris Hodgson Mark Cave-Ayland Norman Vine Carl Anderson Ralph Mason Klaus Foerster Olivier Courtin Bruno Wolff III Markus Schaber Obe Regina Dave Blasby, Sandro Santilli. Postgis, 2009. <http://postgis.refractor.net/>.
- [Fou09] Apache Software Foundation. Apache HTTP server project, 1996-2009. <http://httpd.apache.org/>, accessed June 2009.
- [fS04] CEN European Committee for Standardisation. *Eurocode 8 – Design of Structures for Earthquake Resistance*. BSI, 2004.
- [Gra81] Jim Gray. The transaction concept: Virtues and limitations, 1981. <http://research.microsoft.com/en-us/um/people/gray/papers/theTransactionConcept.pdf>, accessed June 2009.
- [Gro09a] PHP Documentation Group. PHP manual, 2009. <http://www.php.net/manual/en/>, accessed June 2009.

REFERENCES

- [Gro09b] PHP Documentation Group. PHP Manual, GD, 2009. <http://www.php.net/manual/en/book.image.php>, accessed June 2009.
- [Gro09c] PHP Documentation Group. PHP: Odbc - manual, 2009. <http://us2.php.net/manual/en/book.uodbc.php>, accessed on June 2009.
- [Gro09d] PostgreSQL Global Development Group. Postgresql, 2009. <http://www.postgresql.org/>, accessed June 2009.
- [IGC08] I. Iervolino, C. Galasso, and E. Cosenza. Selezione assistita di input sismico e nuove norme tecniche per le costruzioni. proc. of valutazione e riduzione della vulnerabilità sismica di edifici esistenti in c.a. *Convegno ReLUIS*, May 2008. http://www.reluis.it/doc/software/NTC_Records.pdf, accessed June 2009.
- [LLC09] DedaSys LLC. Programming language popularity, 2007-2009. <http://www.langpop.com/>, accessed June 2009.
- [Mat09] Inc. MathWorks. Matlab – the language of technical computing, 2009. <http://www.mathworks.com/products/matlab/>, accessed June 2009.
- [McD03] Tina Spain McDuffie. *Javascript: Concepts & Techniques; Programming Interactive Web Sites*. Franklin, Beedle & Associates, 2003.
- [oC09] U.S. Department of Commerce. National earthquake hazards reduction program, 2009. <http://www.nehrp.gov/>, accessed June 2009.
- [oCE03] American Society of Civil Engineers. *Minimum Design Loads for Buildings & Other Structures, SEI*. American Society of Civil Engineers, 2003.
- [Ree79] Trygve Reenskaug. Thing-model-view-editor. Technical report, Xerox PARC, 1979.
- [Ree03] Trygve Reenskaug. The Model-View-Controller (MVC), its past and present. In *JAOO Conference*, Denmark, 2003.
- [Res09] John Resig. Jquery documentation, 2009. http://docs.jquery.com/Main_Page, accessed June 2009.
- [Sla08] Mark Slagell. Ruby user’s guide, 2008. <http://www.rubyist.net/~slagell/ruby/>, accessed June 2009.
- [Tea09] Rails Core Team. Ruby on Rails, 2009. <http://rubyonrails.org/>, accessed June 2009.
- [W3C02] W3C. Xhtml 1.0 the extensible hypertext markup language (second edition), 2002. <http://www.w3.org/TR/xhtml1/>, accessed June 2009.
- [W3C09] W3C. Cascading style sheets, level 2 revision 1 (css 2.1) specification, 2009. <http://www.w3.org/TR/CSS2/>, accessed June 2009.

Appendix A

American Regulations

According to [oCE03], the spectrum used for seismic analysis contains four sections, each defined by a different equation. Below are shown the different parts of the spectrum according to the Period considered.

- $T < T_0$

$$S_A = S_{DS}(0.4 + 0.6 \frac{T}{T_0}) \quad (\text{A.1})$$

- $T_0 < T < T_S$

$$S_A = S_{DS} \quad (\text{A.2})$$

- $T_S < T < T_L$

$$S_A = \frac{S_{D1}}{T} \quad (\text{A.3})$$

- $T_L < T$

$$S_A = \frac{S_{D1} T_L}{T^2} \quad (\text{A.4})$$

The different values on the equations A.1 – A.4 are given by other formulae and tables that define specific coefficients. In particular the ones from the MCE spectral response acceleration for short periods, S_{MS} , and for a period of 1 second, S_{D1} :

$$S_{MS} = F_a S_S \quad (\text{A.5})$$

$$S_{M1} = F_v S_1 \quad (\text{A.6})$$

Further, the design earthquake spectral response acceleration parameter at short period, S_{DS} , and at a period of 1 second, S_{D1} , are calculated from:

American Regulations

Table A.1: Site Coefficient F_a

Site Class	$S_S < 0.25$	$S_S = 0.5$	$S_S = 0.75$	$S_S = 1.0$	$S_S > 1.25$
A	0.8	0.8	0.8	0.8	0.8
B	1.0	1.0	1.0	1.0	1.0
C	1.2	1.2	1.1	1.0	1.0
D	1.6	1.4	1.2	1.1	1.0
E	2.5	1.7	1.2	0.9	0.9
F	Defined under section 11.4.7 of [oCE03]				

$$S_{DS} = \frac{2}{3} S_{MS} \quad (\text{A.7})$$

$$S_{D1} = \frac{2}{3} S_{M1} \quad (\text{A.8})$$

Therefore, the input the user must give when calculating a D_{rms} following American regulations for the target spectra are as follows:

1. **Site Class** For the coefficient tables
2. S_S The mapped MCE spectral response acceleration at short periods
3. S_1 The mapped MCE spectral response acceleration at a period of 1s
4. T_L Long-period transition period

All this process is described in further detail on the standards book *Minimum Design Loads for Buildings and Other Structures*[oCE03].

Table A.2: Site Coefficient F_v

Site Class	$S_S < 0.1$	$S_S = 0.2$	$S_S = 0.3$	$S_S = 0.4$	$S_S > 0.5$
A	0.8	0.8	0.8	0.8	0.8
B	1.0	1.0	1.0	1.0	1.0
C	1.7	1.6	1.5	1.4	1.3
D	2.4	2.0	1.8	1.6	1.5
E	3.5	3.2	2.8	2.4	2.4
F	Defined under section 11.4.7 of [oCE03]				

Appendix B

European Regulations

The Eurocode 8 [fS04] defines a different set of equations to calculate the desired spectra. Like the American spectra, the European is also divided in several sections, given by:

- $T < T_B$

$$S_e(T) = a_g \cdot S \cdot \left[1 + \frac{T}{T_B} \cdot (\eta \cdot 2,5 \cdot 1)\right] \quad (\text{B.1})$$

- $T_B < T < T_C$

$$S_e(T) = a_g \cdot S \cdot \eta \cdot 2,5 \quad (\text{B.2})$$

- $T_C < T < T_D$

$$S_e(T) = a_g \cdot S \cdot \eta \cdot 2,5 \left[\frac{T_C}{T}\right] \quad (\text{B.3})$$

- $T_D < T < 4\text{s}$

$$S_e(T) = a_g \cdot S \cdot \eta \cdot 2,5 \left[\frac{T_C T_D}{T^2}\right] \quad (\text{B.4})$$

In this case, in order to fully define the spectra, the user must quantify:

1. a_g The design ground acceleration on type A ground
2. T_B The lower limit of the period of the constant spectral acceleration branch
3. T_C The upper limit of the period of the constant spectral acceleration branch
4. T_D The value defining the beginning of the constant displacement response range of the spectrum
5. S The soil factor
6. η The damping correction factor

European Regulations

Table B.1: Type 1 Elastic Response Spectra Parameters

Ground type	S	$T_B(s)$	$T_C(s)$	$T_D(s)$
A	1.00	0.15	0.4	2.0
B	1.20	0.15	0.5	2.0
C	1.15	0.20	0.6	2.0
D	1.35	0.20	0.8	2.0
E	1.40	0.15	0.5	2.0

The values for the periods T_B , T_C , T_D , and the soil factor S depend upon the ground type. Depending on the type of spectra (European regulations contemplate Type 1 and Type 2), a different table of values is used. The two tables are reproduced here in full.

The damping correction factor may be calculated by:

$$\eta = \sqrt{\frac{10}{5 + \xi}} \geq 0,55 \quad (\text{B.5})$$

To successfully identify a spectrum using the european guidelines, a user need only input the following information:

1. a_g – The design ground acceleration on type A ground
2. Type – The spectra type, 1 or 2
3. Ground type – The type of ground present on the site, identified by a letter A-E

All this process is described in further detail on the standards book *Eurocode 8* [fS04].

Table B.2: Type 2 Elastic Response Spectra Parameters

Ground type	S	$T_B(s)$	$T_C(s)$	$T_D(s)$
A	1.00	0.05	0.25	1.2
B	1.35	0.05	0.25	1.2
C	1.50	0.10	0.25	1.2
D	1.80	0.10	0.30	1.2
E	1.60	0.05	0.25	1.2