CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. (2009)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1519

The Collaboratory for the Study of Earthquake Predictability perspective on computational earthquake science

J. Douglas Zechar[1,*,†], Danijel Schorlemmer[1,2], Maria Liukis[2], John Yu[1,2], Fabian Euchner[3], Philip J. Maechling[2] and Thomas H. Jordan[1,2]

[1]*Department of Earth Sciences, University of Southern California, Los Angeles, CA 90089, U.S.A.*
[2]*Southern California Earthquake Center, University of Southern California, Los Angeles, CA 90089, U.S.A.*
[3]*Swiss Seismological Service, ETH Zürich, Sonneggstrasse 5, 8092 Zürich, Switzerland*

SUMMARY

The Collaboratory for the Study of Earthquake Predictability (CSEP) aims to advance earthquake research by rigorous testing of earthquake forecast hypotheses. As in other disciplines, such hypothesis testing requires carefully designed experiments that meet certain requirements: they should be reproducible, fully transparent, and conducted within a controlled environment. CSEP has begun building infrastructure for conducting such rigorous earthquake forecasting experiments. Because past earthquake prediction experiments often have been controversial, CSEP testing centers—the secure, controlled computational environments within which experiments are conducted—have been designed to address particular issues related to transparency and exact reproducibility. Moreover, CSEP fosters collaboration among scientists developing earthquake forecast models, and the testing center concept allows multiple concurrent predictability experiments. In this paper, we share our perspective on computational earthquake science by presenting the design principles, organizational structure, and implementation details of CSEP testing centers. We describe ongoing forecast experiments in different testing regions and some of the

*Correspondence to: J. Douglas Zechar, Lamont-Doherty Earth Observatory of Columbia University, 61 Route 9W, Palisades, NY 10964, U.S.A.
†E-mail: zechar@usc.edu, zechar@Ldeo.columbia.edu

Contract/grant sponsor: W. M. Keck Foundation
Contract/grant sponsor: Southern California Earthquake Center
Contract/grant sponsor: NSF Cooperative Agreement; contract/grant number: EAR-0106924
Contract/grant sponsor: USGS Cooperative Agreement; contract/grant number: 02HQAG0008

implementation challenges encountered. We also describe the collaboration tools used for multinational software development and regional presentation websites. The need for common data exchange formats is discussed, as are potential avenues of future research within CSEP testing centers. Copyright © 2009 John Wiley & Sons, Ltd.

Received 9 October 2008; Accepted 24 July 2009

KEY WORDS: earthquake predictability; transparency; reproducibility

1. INTRODUCTION

Throughout the past several decades, individual investigators and small research groups have concentrated efforts on *operational earthquake prediction*: the ability to forecast damaging earthquakes far enough in advance to mitigate losses and avoid casualties. The majority of the resulting studies can be characterized as a search for physical precursors to large earthquakes; often suggested in retrospect—that is, signals were deemed precursory only after a large earthquake occurred—no such precursor has proved reliable. The Collaboratory for the Study of Earthquake Predictability (CSEP), a large international community of geoscientists, statisticians, and computational scientists, takes a more fundamental approach to the problems of earthquake prediction by seeking to identify and understand predictable characteristics of the earthquake system.

CSEP aims to reduce the controversy that often obscures earthquake prediction experiments by applying a few key principles: predictions are expressed in precise and comparable formats; data used to create and evaluate the forecasts are well-defined; forecasts are evaluated using well-defined testing procedures; and forecast generation and evaluation are conducted in controlled computational environments called testing centers [1]. Hypothesis testing is central to CSEP research and particular attention is paid to precise experiment specification and rigorous testing of prospective earthquake forecasts. As such, CSEP has a unique role in computational earthquake science: to provide infrastructure for researchers to develop, test, and compare models of earthquake occurrence. On a day-to-day basis, this system must interact with multiple complex forecast models, collect observations, execute forecast evaluation tests, and generate accessible experiment results. Moreover, CSEP testing centers must accommodate new models, new experiments, and new evaluation procedures. Some of the challenges of designing, implementing, and operating the CSEP system are common to large scientific collaborations, and some are of particular interest to members of the computational earthquake science community.

In this paper, we provide a detailed description of the CSEP computational infrastructure and its relationship to the scientific questions of interest. We discuss general CSEP concepts as well as specific implementation details of the W. M. Keck CSEP Testing Center at the Southern California Earthquake Center (SCEC), which has been in operation for more than two years. We briefly describe some of the experiments that are currently underway within CSEP testing centers and delineate our design principles and implementation strategies. To illustrate system functionality, we describe a typical month of operations in a testing center. We discuss advantages of our open source development approach and describe potential future interactions between CSEP and members of the ACES working group.

DOI: 10.1002/cpe

2. SCIENCE OF CSEP

CSEP seeks to quantify the predictability of the earthquake rupture process. CSEP scientists currently approach this problem by developing multiple earthquake forecast models and conducting controlled scientific experiments to compare models with each other and with observations. Complementary to these efforts are the development of testing methods and the characterization of performance metrics. This work continues because it is not obvious how scientific earthquake predictions should be evaluated, and each performance metric emphasizes a different aspect of the prediction problem. CSEP experiments are conducted within *testing centers*—based on a concept first presented by Schorlemmer and Gerstenberger [1]—and the geographic study area for a given experiment is called the *testing region*.

The W. M. Keck Testing Center at SCEC currently hosts a number of forecast experiments in different testing regions. The experiment proposed by the Regional Earthquake Likelihood Models (RELM) working group is underway [2,3]; this is a five-year experiment to compare more than a dozen competing models that forecast the number of earthquakes with magnitude $M \geq 4.95$ in latitude/longitude/magnitude bins in and around California [4–11]. Two models that issue daily forecasts of the number of earthquakes with magnitude $M \geq 3.95$ in and around California are also under test [12,13], as are eight models that issue forecasts for three months of seismicity [14]. In two Western Pacific testing regions, three models provide annual forecasts of events with moment magnitude $M_W \geq 5.8$ and one model provides daily forecasts of the same [15,16]. All of these experiments are conducted prospectively; that is, the models are based only on information available prior to issuing the forecast.

The scientific goals of CSEP drive the computational developments of the testing centers. For example, CSEP testing centers are designed with an emphasis on integrating and comparing multiple heterogeneous models and data streams, and these goals require that testing centers be appropriately flexible. Because experiment evaluation is fundamentally important to the scientific pursuits of CSEP, testing methods are carefully engineered and optimized. The desire to conduct experiments in multiple testing regions also influences the design of the testing center, necessitating a modular development approach that allows the addition of new testing regions with new data streams and new evaluation methods. CSEP testing centers must accommodate new experiments and new performance metrics while maintaining a stable environment. In the following section, we mention some of the challenges related to designing testing centers and describe the organization and guiding principles of CSEP computational development.

3. ORGANIZATION AND PRINCIPLES

Given the scientific emphasis on experimentation, CSEP is organized in geographically separated computational environments that host specific experiments. Currently, testing centers are operational in Los Angeles (SCEC), Wellington (GNS Science), and Zurich (Swiss Federal Institute of Technology; ETH); another is under development in Tokyo (Earthquake Research Institute (ERI)). Predictability experiments are grouped by the testing region to which they are applied, with the region specified by physical boundaries in latitude, longitude, and depth. A testing center can host experiments in multiple testing regions, and multiple testing centers can host experiments in the

DOI: 10.1002/cpe

same testing region. At present, the SCEC testing center hosts experiments for testing regions in the Northwest Pacific, Southwest Pacific, and California; the GNS Science testing center hosts experiments for New Zealand; and the ETH testing center hosts experiments for a testing region in Italy. Within each testing region, experiments are further classified into forecast classes. A forecast class contains one or more forecast models that participate in the same experiment, and therefore models belonging to the same forecast class are directly comparable. For example, as mentioned in Section 2, the California testing region contains a one-day forecast class consisting of two models that issue daily earthquake forecasts.

Although this organizational structure addresses the scientific and collaborative goals of CSEP, it presents some challenges. Each predictability experiment is unique, requiring specific data streams and/or processing of those data. For example, experiments in the two Western Pacific testing regions use the Global Centroid Moment Tensor earthquake catalog [17,18] while experiments in California use the Advanced National Seismic System (ANSS) catalog [19]; an additional complication arises because a subset of the experiments in California require catalog declustering. Most of the problems caused by this experiment-level complexity have been addressed by adopting an object-oriented software development approach (described in the Section 4 below). On the other hand, system-level challenges such as managing multiple testing centers has proven rather more difficult; the majority of CSEP software development occurs at SCEC, and updating remote testing centers typically requires time, a physical presence at the remote center, and local expertise. Currently, the problem of keeping testing centers automatically up-to-date using the latest version of testing center codes is being addressed via Yellowdog Updater, Modified (YUM) repository distribution, but this topic is still under investigation.

Transparency and reproducibility are the two most important principles guiding the conduct of CSEP experiments. If experiment procedures are transparent and results are reproducible, researchers can concentrate their efforts on interpretation and physical understanding. Therefore, when possible, CSEP takes an open approach that is characterized by choosing reliable open source packages, making data freely accessible, and sharing internally developed testing center software with all participants. Additionally, great care is taken to ensure exact reproducibility of experiment results. Earthquake catalogs change many times each day; therefore, to reproduce the results of an experiment using catalog data, one must use the same catalog data employed in the initial experiment. By necessity, CSEP therefore maintains a copy of every catalog used in every experiment. A more extreme practice involves random number generation: if a model uses random numbers to generate a forecast, the seed of the random number generator is persisted to storage so that any exact sequence can be reproduced. There is also considerable infrastructure in place to automatically rerun a specific experiment or batch of experiments; this process is aided considerably by metadata that is created and saved whenever an experiment is run.

Prospective forecast experiments may take several years to observe significantly large samples, particularly when dealing with large target earthquakes. Therefore, CSEP testing centers are designed to run for many years without great maintenance efforts or expense, relying on few commercial packages and a modest set of machines. To maintain system stability, changes to a CSEP testing center are applied according to a schedule that includes ample time for testing and troubleshooting, rather than applying changes on a whim. Along these same lines, CSEP testing centers include an environment that mirrors, but does not interfere with, the day-to-day operational system (see Section 5); this configuration enables ongoing development and integration of new models.

DOI: 10.1002/cpe

4. IMPLEMENTATION

The implementation details of a CSEP testing center are perhaps best understood in the context of normal, day-to-day operations. Table I lists experiment activities that typically occur each month in the operational SCEC testing center. This list is not exhaustive as it does not include, for example, ongoing development and debugging of new model codes.

The computational core of the CSEP testing center is called the *dispatcher*, the central processing module that performs the following tasks in an end-to-end fashion: retrieval and processing of earthquake catalog observation data, forecast model invocation (used to generate forecasts), forecast evaluation, and publication of experiment results (see Figure 1). Catalog retrieval and processing begins with downloading the catalog from the appropriate agency, storing the catalog files with metadata, and filtering the catalog so that only the events of interest for the given experiment remain. The one-day and three-month model codes use the catalog as input, and all experiments require a catalog for evaluation. Simulations are used to account for measurement uncertainties in the observed target earthquake catalog, resulting in a suite of modified observation catalogs. For one-day models, the specific model codes are executed to generate forecasts; for other models, the forecasts are simply scaled to the testing period. For example, when a model belonging to the five-year RELM mainshock-aftershock forecast group is evaluated at the end of one year, the initial forecast values are divided by five. The resulting forecast is then evaluated using tests described

Table I. Experiment activities for a typical month in the SCEC testing center.

Occurs	Testing region	Forecast group	Task	Approximate execution time
Daily	CA	CA one-day	Generate 3 forecasts	1 h 50 min
Daily	CA	CA one-day	Evaluate 3 forecasts ($x5$)	1 h 15 min
Daily	NWP	NWP one-day	Generate 1 forecast	1.5 min
Daily	NWP	NWP one-day	Evaluate 1 forecast ($x4$)	2.5 min
Daily	SWP	SWP one-day	Generate 1 forecast	2 min
Daily	SWP	SWP one-day	Evaluate 1 forecast ($x4$)	2.5 min
Monthly	CA	RELM mainshock-aftershock	Evaluate 5 forecasts ($x7$)	3 h
Monthly	CA	RELM mainshock-aftershock.2	Evaluate 6 forecasts ($x7$)	3 h
Monthly	CA	RELM mainshock	Evaluate 12 forecasts ($x7$)	11 h 30 min
Monthly	CA	RELM mainshock.2	Evaluate 13 forecasts ($x7$)	8 h
Monthly	CA	CA five-year alarm-based	Evaluate 1 forecast ($x2$)	15 s
Monthly	CA	CA five-year rate-based	Evaluate 2 forecasts ($x7$)	2 h 25 min
Semi-monthly	CA	CA three-month	Evaluate 7 forecasts ($x5$)	13 h
Semi-monthly	NWP	NWP one-year alarm-based	Evaluate 1forecast ($x1$)	15 s
Semi-monthly	NWP	NWP one-year rate-based	Evaluate 3forecasts ($x7$)	53 min
Semi-monthly	SWP	SWP one-year alarm-based	Evaluate 1forecast ($x1$)	10 s
Semi-monthly	SWP	SWP one-year rate-based	Evaluate 3 forecasts ($x7$)	54 min

Testing region abbreviations—*CA*: California, *NWP*: Northwest Pacific, *SWP*: Southwest Pacific.

In the *Task* column, the number in parentheses indicates the number of testing metrics computed for each forecast. *Approximate execution time* data include obtaining and processing data stream; RELM forecast groups have longer execution time than RELM.2, because catalog processing and simulation are done by RELM group and results are used by both forecast groups.

DOI: 10.1002/cpe

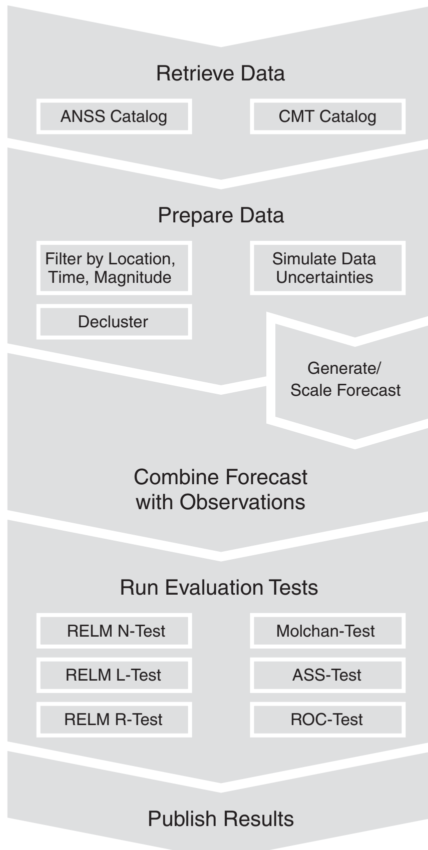

Figure 1. Schematic of dispatcher workflow for a typical predictability experiment in a CSEP testing center.

by Schorlemmer *et al.* [20] and Zechar and Jordan [21,22]. The results of these tests are captured with metadata, and plots of the results are published to a web server for presentation.

For daily processing such as the tasks listed in Table I, the dispatcher is scheduled to run automatically via cron jobs. Alternatively, the dispatcher can be invoked directly for research uses such as running or re-running a specific experiment or applying a new evaluation metric to a previous forecast. In all processing steps, the dispatcher interacts with other scientific codes written in a variety of programming languages—Fortran, MATLAB, Java, R, and Python are all used. The dispatcher itself is written in Python, which was chosen for internal testing center development for several reasons: it is free and open source; it is object-oriented by design; it has powerful built-in types that speed new development; it provides an automated testing framework; and it provides interoperability that makes it an ideal tool for interacting with other codes.

In addition to transparent end-to-end processing, the other key functionality of the dispatcher is the ability to reproduce exactly a specific experiment. To repeat an experiment, the dispatcher relies heavily on metadata. Experiment metadata describe the locations of the forecast or forecast model

DOI: 10.1002/cpe

codes, input catalogs, and observation catalogs; command line options such as the evaluation metric employed; and random number generator seeds. Once the dispatcher has identified these properties, a copy of each is made to preserve the integrity of the original experiment data and the experiment can be repeated.

5. TOPOLOGY AND TOOLS USED BY TESTING CENTER

CSEP testing centers use a configuration with four machines: three machines with identical hardware are used for experiments and new developments, and a fourth hosts the results and the testing center website (see Figure 2). The three primary machines are referred to as the development, certification, and operational environments, respectively. This architecture was designed to maximize system stability and to isolate the operational environment from the rapidly-changing development system. The development machine is used for implementing new functionality and debugging new forecast models. When it appears that a new model or new code is ready for testing, it is moved to the certification machine and checked for stability and consistency over a period of weeks to months.

System integration and verification are done with automated build and acceptance tests using the CruiseControl framework (http://cruisecontrol.sourceforge.net/); any errors are logged and sent to the development team. The build process typically involves compiling forecast model codes and performing a system health check to verify that no conflicts have been created. System builds are archived to allow reproducibility, and CruiseControl runs daily to reduce integration problems.

CSEP development makes extensive use of automated acceptance testing. These automated test suites ensure that new changes work properly and that they do not cause conflicts with existing system capabilities. Acceptance tests are individually designed for each new forecast model, evaluation metric, data stream, or data processing step, but the tests derive from standard base classes and are therefore easy to implement. For example, when a modeler wants to submit forecast codes to an existing forecast class, he/she must also supply example input data and the corresponding expected output. In this case, the acceptance test is as simple as verifying that the code yields the expected output when executed in the certification environment; in certain cases, a specified tolerance for difference is used, for example, when using a different compiler from the one used in the modeler's environment.

Also within the certification environment, tasks such as those listed in Table I are automatically executed to mirror the operational environment; this is particularly important just prior to a new release because it allows for extensive testing of all the functionality of the operational system. The certification environment provides a standard software stack: Python, MATLAB, Java, Message Passing Interface CHameleon (MPICH), R, g77, and gfortran are available for forecast model and evaluation metric use, as are the dispatcher and auxiliary gcc, Linux, and Python codes. The operational environment hosts the same software stack and is used solely for scientific testing; experiment results produced within the operational system are published to the web environment. The main CSEP website can be found at http://www.cseptesting.org and individual testing centers are linked from there. The current testing center websites use the Drupal content management system (http://drupal.org) and a custom results viewer developed in PHP. An example of the results viewer interface is shown in Figure 3. The results viewer provides access to performance metric plots, including their evolution through time as experiments progress.

DOI: 10.1002/cpe

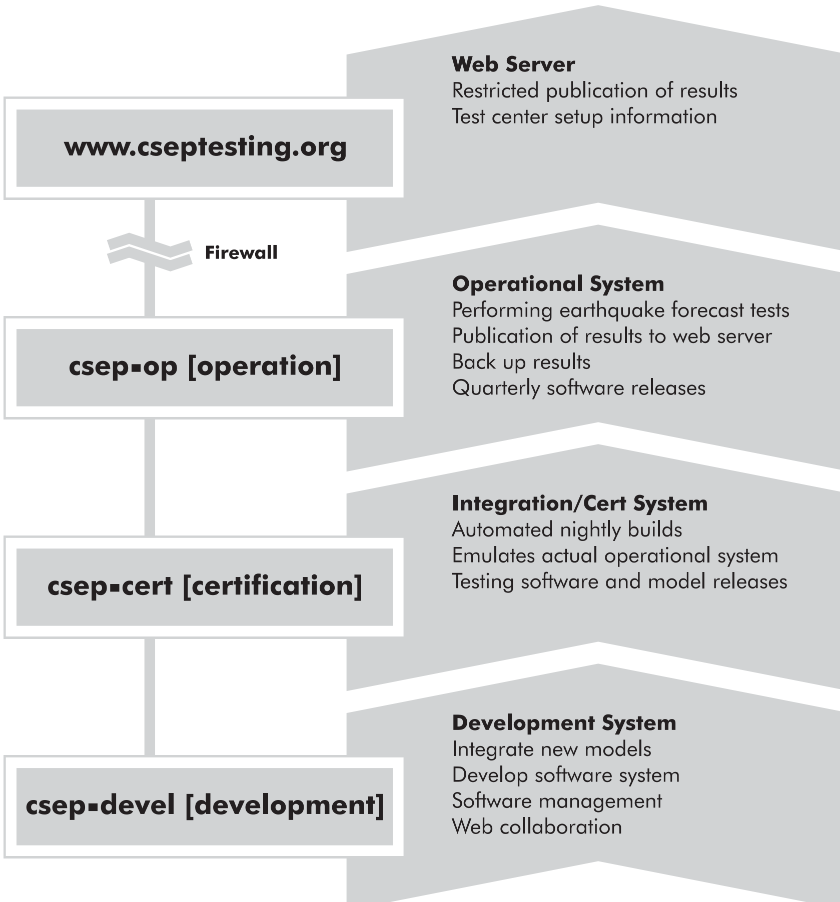

Figure 2. Diagram of CSEP testing center four-machine layout depicting interaction between the different environments.

In all aspects of testing center development, the open source Subversion revision-control system is used (http://subversion.tigris.org). Use of Subversion provides version control of all codes and permits documentation of revisions. Each major release of the testing center software is tagged, making it possible to revisit any previous version for reproducing previous experiments. The CSEP development team also uses the open source Trac package (http://trac.edgewall.org/) as a web collaboration platform and issue tracking system, where bugs and system enhancement requests are documented.

Following the principles of transparency and low maintenance costs, the Linux operating system is used on all machines in CSEP testing centers. Currently in the SCEC testing center, each machine runs the 64-bit version of Fedora Core 10 on systems with two quad-core Intel Xeon E5420 processors at 2.5 Ghz and 12 GB RAM. Storage in the SCEC testing center totals 7.2 TB of

DOI: 10.1002/cpe

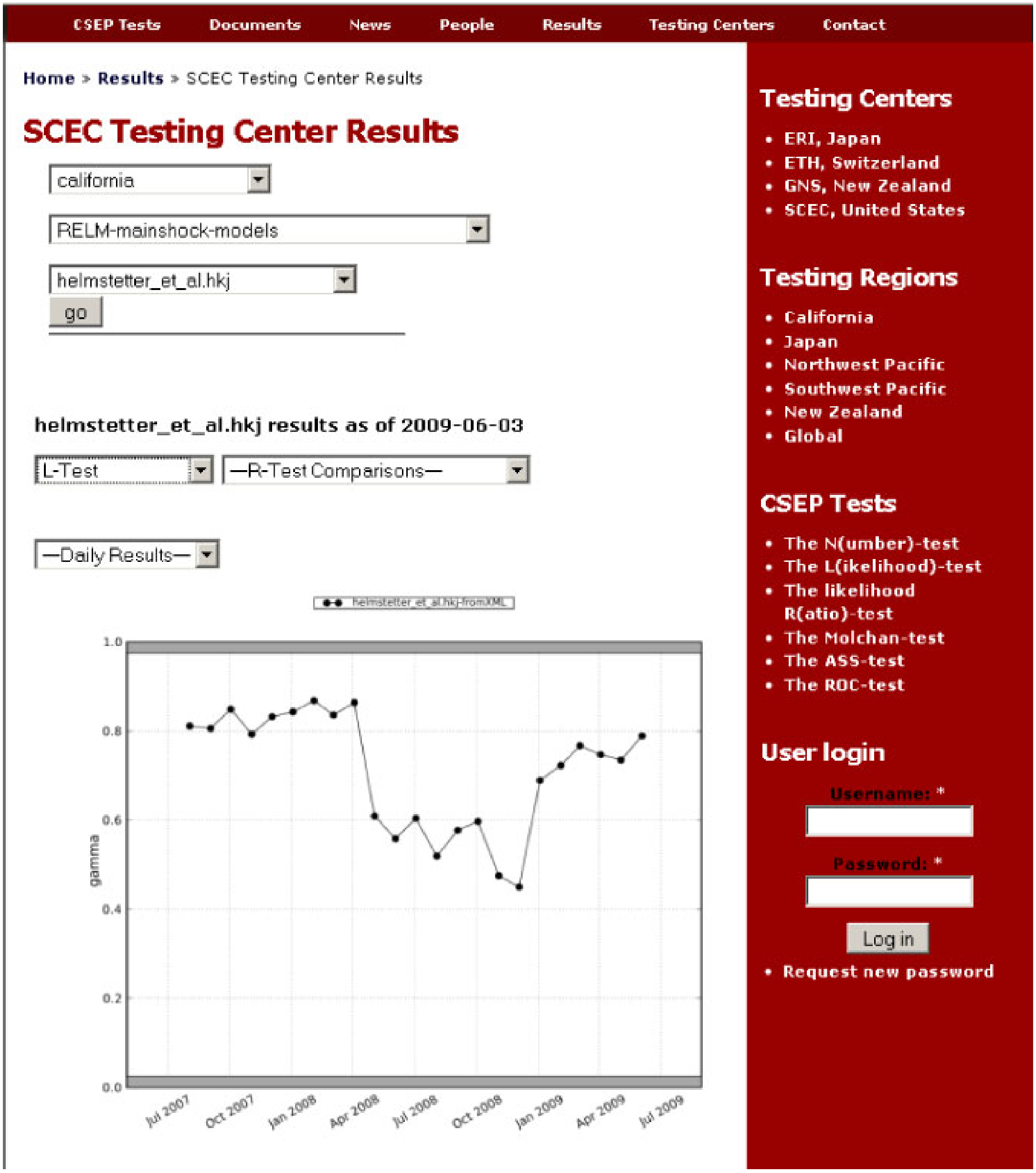

Figure 3. Example results from SCEC testing center, CA RELM-mainshock forecast class. Here, an example evaluation metric and its time-varying behavior are also shown.

DOI: 10.1002/cpe

NFS-mounted drives in a RAID 5 configuration. To increase system stability, standard Fedora Linux packages are maintained in a local YUM repository and nonstandard packages are maintained in a separate Subversion repository; the certification and operational machines make quarterly software updates from these repositories only. Owing to such infrequent updates, these machines are behind a firewall.

6. DISCUSSION

Despite addressing most of the scientific and collaborative goals of CSEP, the testing center infrastructure requires additional development. One of the major outstanding challenges is the ability to automatically distribute the functionality developed at SCEC to other centers. Related to this issue is the problem of exporting functionality from one testing center and configuring it to work in a different computational environment. This is a problem of both system details and scale—for example, it may be desirable for an individual researcher to establish a mini-testing center on his laptop, or another CSEP institution may use different hardware. It seems that there are no easy solutions to these problems but rather they must be dealt with on a case-by-case basis.

Another area for development is the delivery of CSEP data products. Researchers have expressed interest in accessing intermediate data sets such as declustered catalogs, produced by the dispatcher during routine processing. Scientists whose models are being evaluated are also interested in obtaining detailed forecast results, including log files and intermediate files produced by their models. Currently, it is difficult for researchers to access these data sets from the controlled operational CSEP testing environment. To address this, we plan to develop data access interfaces for obtaining intermediate data from operational testing centers. Through these interfaces, experimenters could work on any phase of CSEP forecast testing without needing to run a complete end-to-end experiment.

It seems that the major development bottlenecks are caused by a lack of community standards regarding data formats. The CSEP infrastructure exchanges data internally in both XML and legacy formats; for example, the ANSS raw catalog data is converted to the legacy ZMAP format [23], but forecasts and results are manipulated in XML formats that are loosely based on the developing QuakeML standard [24] (http://www.quakeml.org), which will eventually replace internal legacy formats. Now that the testing center software is in stable operation, integration of new forecast models is perhaps the most time-consuming aspect of development. In particular, each model code seems to use its own custom formats for input data and the resulting output forecasts. Some scientists are willing to provide codes that conform to the standard formats used by CSEP but, if not, this requires custom development of translation services. The collaborative nature of CSEP should naturally encourage modelers to begin using standard formats as it will speed up integration of multiple models and allow for wider comparison.

While CSEP testing centers currently are used exclusively for prospective testing of earthquake forecasts, the developed infrastructure is flexible enough to support a wide range of collaborative earthquake research efforts. For example, this infrastructure could be used to host ACES work, acting as a repository for the earthquake simulators described elsewhere in this issue. Simulation results could be archived and reproduced exactly, and standards could be developed—deriving from

DOI: 10.1002/cpe

the existing CSEP standard exchange formats—to aid in the comparison of different simulator methods. Although the current implementation of CSEP testing centers is rather specific to the goal of conducting earthquake forecast experiments, the guiding principles and motivations are common to much of computational earthquake science.

ACKNOWLEDGEMENTS

This research was supported by the SCEC. SCEC is funded by NSF Cooperative Agreement EAR-0106924 and USGS Cooperative Agreement 02HQAG0008. The SCEC contribution number for this paper is 1226. The W. M. Keck Collaboratory for Study of Earthquake Predictability Testing Center at SCEC was made possible by the generous financial support of the W. M. Keck Foundation.

REFERENCES

1. Schorlemmer D, Gerstenberger M. RELM Testing Center. *Seismology Research Letters* 2007; **78**(1):30–36. DOI: 10.1785/gssrl.78.1.30.
2. Field E. Overview of the working group for the development of regional earthquake likelihood models. *Seismology Research Letters* 2007; **78**(1):7–16. DOI: 10.1785/gssrl.78.1.7
3. Schorlemmer D, Zechar J, Werner M, Field E, Jackson D, Jordan T. First results of the Regional Earthquake Likelihood Models experiment. *Pure and Applied Geophysics*, 2009; accepted.
4. Bird P, Liu Z. Seismic hazard inferred from tectonics: California. *Seismology Research Letters* 2007; **78**(1):37–48. DOI: 10.1785/gssrl.78.1.37.
5. Ebel J, Chambers D, Kafka A, Baglivo J. Non-Poissonian earthquake clustering and the hidden Markov model as bases for earthquake forecasting in california. *Seismology Research Letters* 2007; **78**(1):57–65. DOI: 10.1785/gssrl.78.1.57.
6. Helmstetter A, Kagan Y, Jackson D. High-resolution time-independent grid-based forecast for $M>5$ earthquakes in California. *Seismology Research Letters* 2007; **78**(1):78–86. DOI: 10.1785/gssrl.78.1.78.
7. Holliday J, Chen C, Tiampo K, Rundle J, Turcotte D, Donnellan A. A RELM earthquake forecast based on pattern informatics. *Seismology Research Letters* 2007; **78**(1):87–93. DOI: 10.1785/gssrl.78.1.87.
8. Kagan Y, Jackson D, Rong Y. A testable five-year forecast of moderate & large earthquakes in southern California based on smoothed seismicity. *Seismology Research Letters* 2007; **78**(1):94–98. DOI: 10.1785/gssrl.78.1.94.
9. Shen Z, Jackson D, Kagan Y. Implications of geodetic strain rate for future earthquakes, with a five-year forecast of M5 earthquakes in southern California. *Seismology Research Letters* 2007; **78**(1):116–120. DOI: 10.1785/gssrl.78.1.116.
10. Ward S. Methods for evaluating earthquake potential and likelihood in and around California. *Seismology Research Letters* 2007; **78**(1):121–133. DOI: 10.1785/gssrl.78.1.121.
11. Wiemer S, Schorlemmer D. ALM: An asperity-based likelihood model for California. *Seismology Research Letters* 2007; **78**(1):134–140. DOI: 10.1785/gssrl.78.1.134.
12. Gerstenberger M, Jones L, Wiemer S. Short-term aftershock probabilities: case studies in California. *Seismology Research Letters* 2007; **78**(1):66–77. DOI: 10.1785/gssrl.78.1.66.
13. Zhuang J, Christophersen A, Savage M, Vere-Jones D, Ogata Y, Jackson D. Differences between spontaneous and triggered earthquakes: Their influences on foreshock probabilities. *Journal of Geophysical Research* 2008; **113**:B11302. DOI: 10.1029/2008JB005579.
14. Rhoades D. Application of the EEPAS model to forecasting earthquakes of moderate magnitude in southern California. *Seismology Research Letters* 2007; **78**(1):110–115. DOI: 10.1785/gssrl.78.1.110.
15. Kagan Y, Jackson D. Probabilistic forecasting of earthquakes. *Geophysics Journal International* 2000; **143**:438–453.
16. Marzocchi W, Lombardi A. A double branching model for earthquake occurrence. *Journal of Geophysical Research* 2008; **113**:B08317. DOI: 10.1029/2007JB005472.
17. Dziewonski A, Chou T-A, Woodhouse J. Determination of earthquake source parameters from waveform data for studies of global and regional seismicity. *Journal of Geophysical Research* 1981; **86**:2825–2852.
18. Ekström G, Dziewonski A, Maternovskaya N, Nettles M. Global seismicity of 2003: Centroid–moment–tensor solutions for 1087 earthquakes. *Physics of the Earth and Planetary Interiors* 2005; **148**(2–4):327–351.
19. ANSS earthquake catalog. Available at: http://quake.geo.berkeley.edu/anss [24 July 2009].
20. Schorlemmer D, Gerstenberger M, Wiemer S, Jackson D, Rhoades D. Earthquake likelihood model testing. *Seismology Research Letters* 2007; **78**(1):17–29. DOI: 10.1785/gssrl.78.1.17.

21. Zechar J, Jordan T. Testing alarm-based earthquake predictions. *Geophysical Journal International* 2008; **172**(2):715–724. DOI: 10.1111/j.1365-246X.2007.03676.x.
22. Zechar J, Jordan T. The area skill score statistic for evaluating earthquake predictability experiments. *Pure and Applied Geophysics* 2009; accepted.
23. Wiemer S. A software package to analyze seismicity: ZMAP. *Seismology Research Letters* 2001; **72**:373–382.
24. Schorlemmer D, Wyss A, Maraini S, Wiemer S, Baer M. QuakeML: An XML schema for seismology. *ORFEUS Newsletter* 2004; **6**(2):9.