The MITRE Elastic Goal-Directed Simulation Framework (MEG)

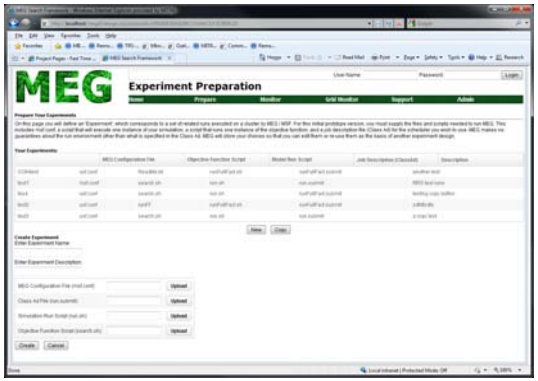
Christine Harvey
The MITRE Corporation

18 June 2013
OSDC Workshop
Edinburgh, UK

MITRE

The MITRE Elastic Goal-Directed Simulation Framework (MEG)

- **Middleware framework to supplement existing simulation applications**
- **Provides access to three capabilities:**
  - Cloud-based or grid-based computing resources
  - Advanced Design of Experiments (DOE) methodologies
  - Robust data processing and visualization tools

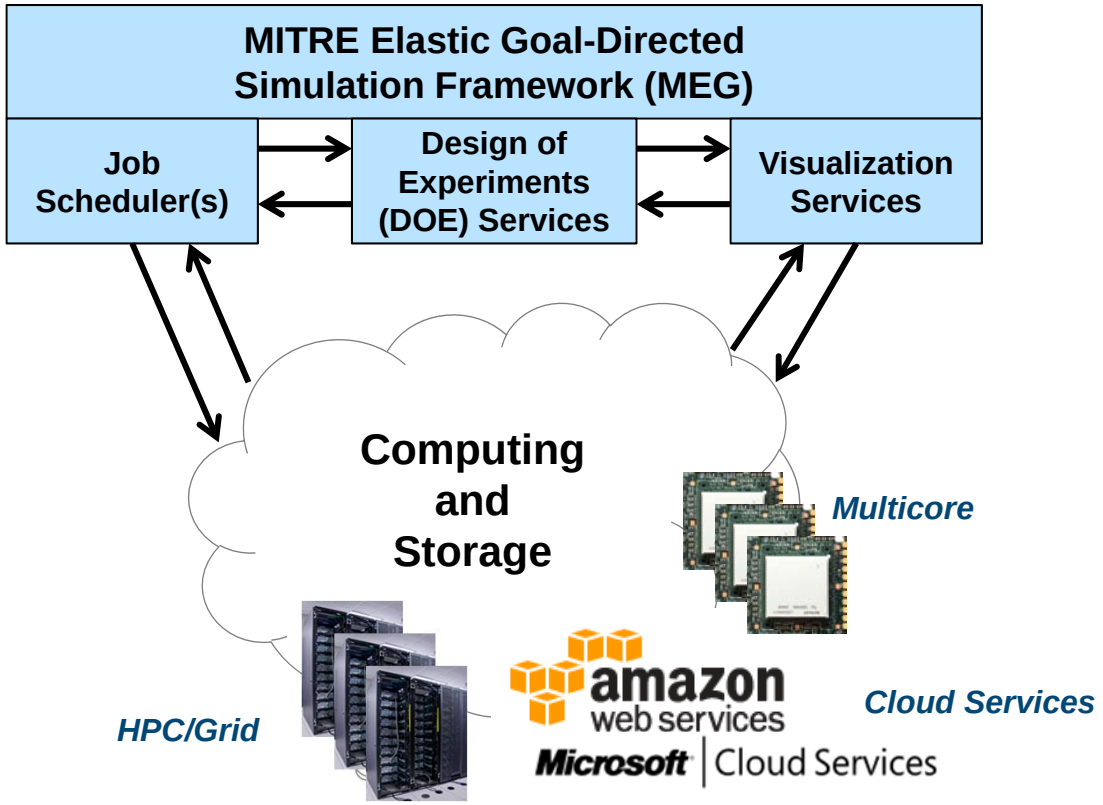


User Interface		
Scheduler Services	**DOE Services**	**Visualization Services**
Data Services		

MITRE

The MITRE Elastic Goal-Directed Simulation Framework (MEG)

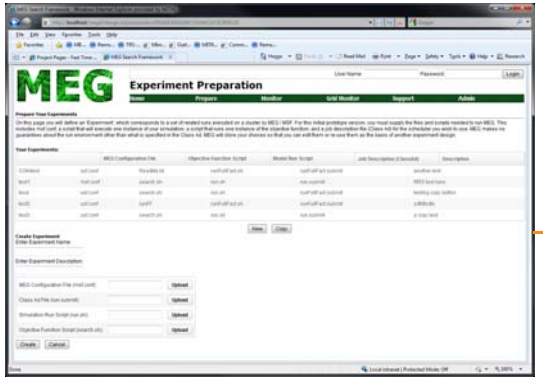
- **Run experiments on multiple grids**
- **Simulation Adaption:**
  - Command line input
  - No hard coded paths



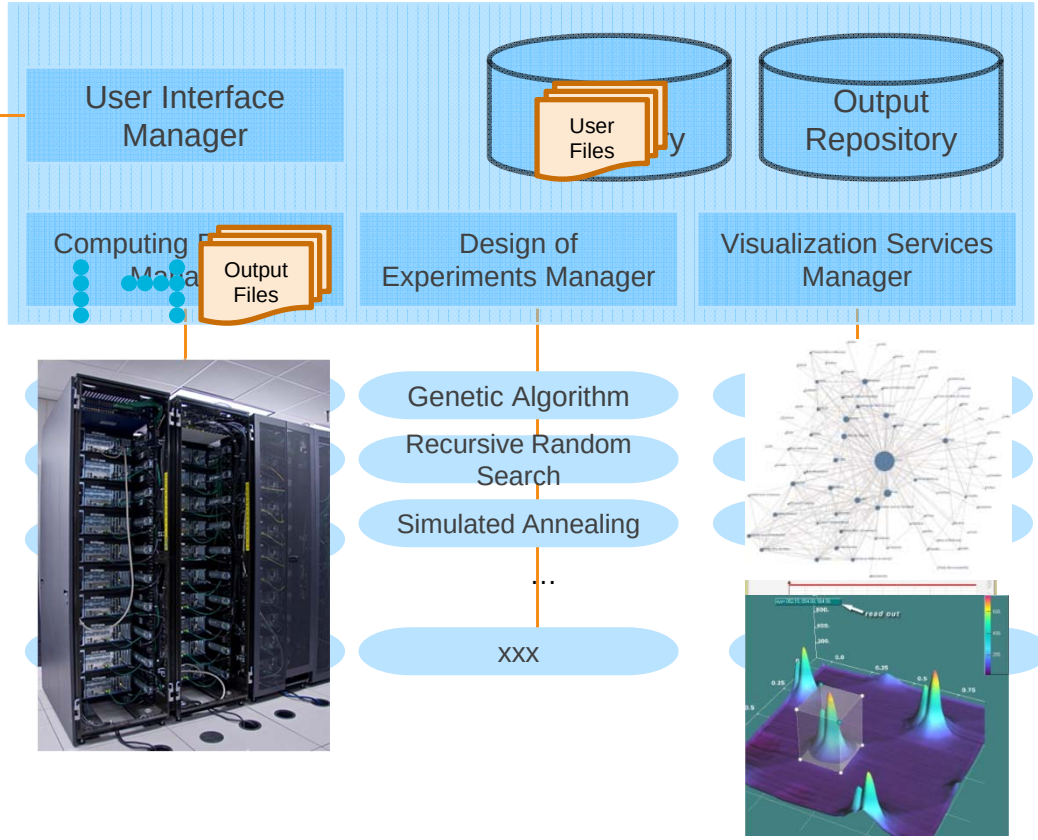
The MITRE Elastic Goal-Directed Simulation Framework (MEG) is first and foremost an engineering activity – our goal is to develop a practical, useful tool.

MITRE

The MEG Vision



Automated Goal-Directed Replication Management Framework



Log into framework web site

Stage files

Specify the Design of Experiments (DOE)

View status of available clusters/grids

Select target cluster/grid

Submit DOE for running

- Files transferred to target cluster
- As jobs complete…
  - Output data is returned
  - DOE Mgr determines next jobs
  - Vis Mgr is updated

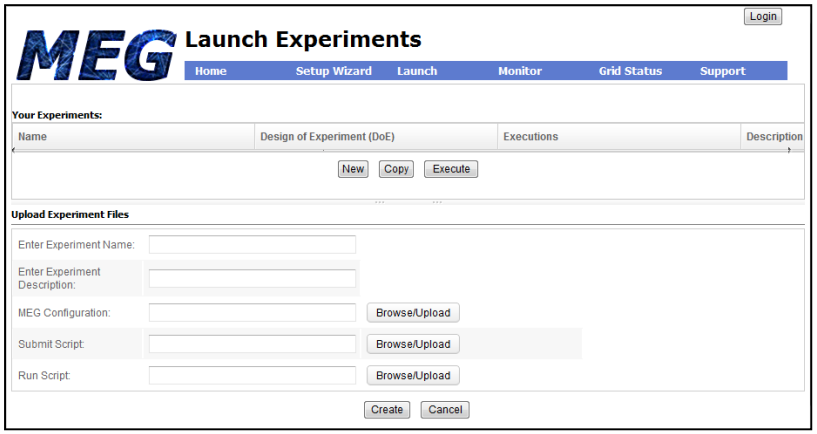
MITRE

Design Principles

- **Learn by Doing –** Give users access early and often. Don't design in a vacuum
- **Provide Transparency –** Let end users see the reason for faults
- **Support Various Modeling Languages –** Don't force the users to change, adapt to their needs
- **Provide a Low Barrier for Entry –** Require the minimum amount of effort for end users to integrate
- **A Good Idea Applies to Itself –** Use simulation-based optimization to identify the optimal configuration for MEG installations

MEG Architecture

- **GUI Services**
  - ZK GUI Toolkit
  - TOMCAT web server
  - Wizard utility
  - Persistent workspace
- **Scheduler Services**
  - Distributed Resource Management by Gridway
  - User does not need to specify grid
- **DOE Services**
  - Parameter sweep, user specified input, Monte Carlo, multiple types of Genetic Algorithms
- **Data and Visualization Services**
  - Output directory is monitored for changes by the MITRE Data Gin
  - Data is synchronized to the database
  - Can be visualized in “fast time”



MITRE

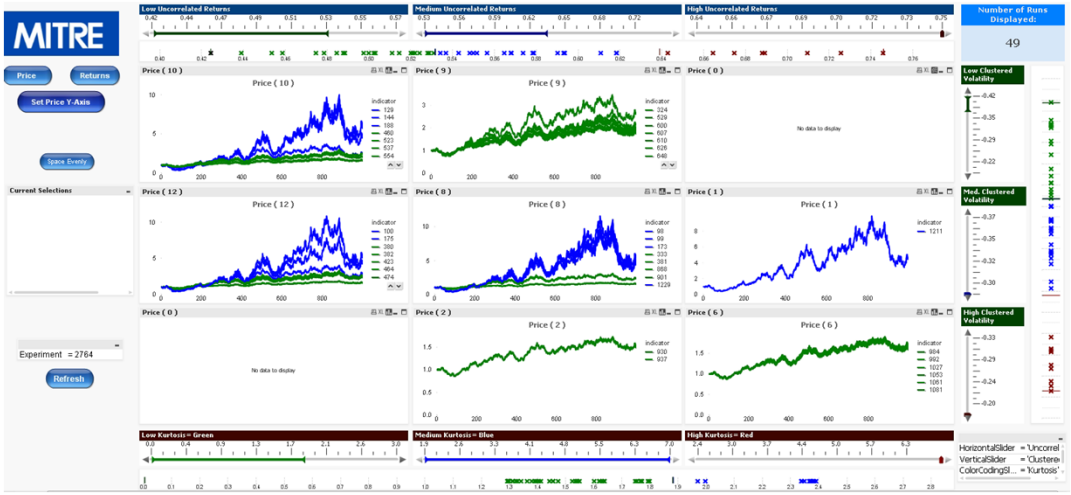
Case Study: Financial Market Model

- Approach to running simulations on the MEG
- Demo

MITRE

Case Study: Financial Market Model

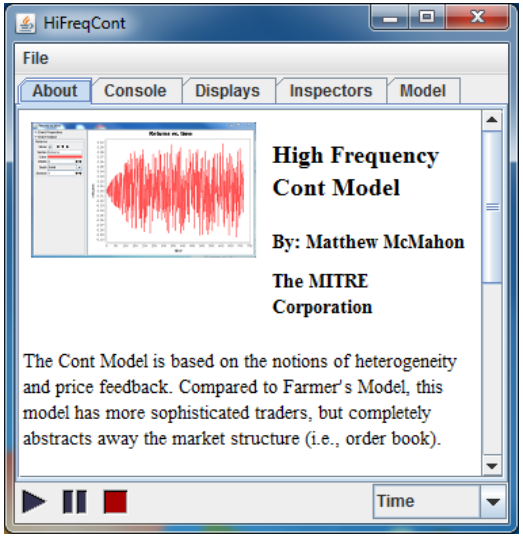
- **Involves two MITRE Innovation Program (MIP) research projects**
  - Financial Modeling & Simulation Execution Environment
    - Rajani Shenoy, Matthew McMahon, Jenny McFarland, Ernie Page
  - Computational Steering for Interactive Modeling and Simulation
    - Carlos Ramos, Matthew McMahon, Thom DeCarlo III



MITRE

Case Study: Financial Market Model

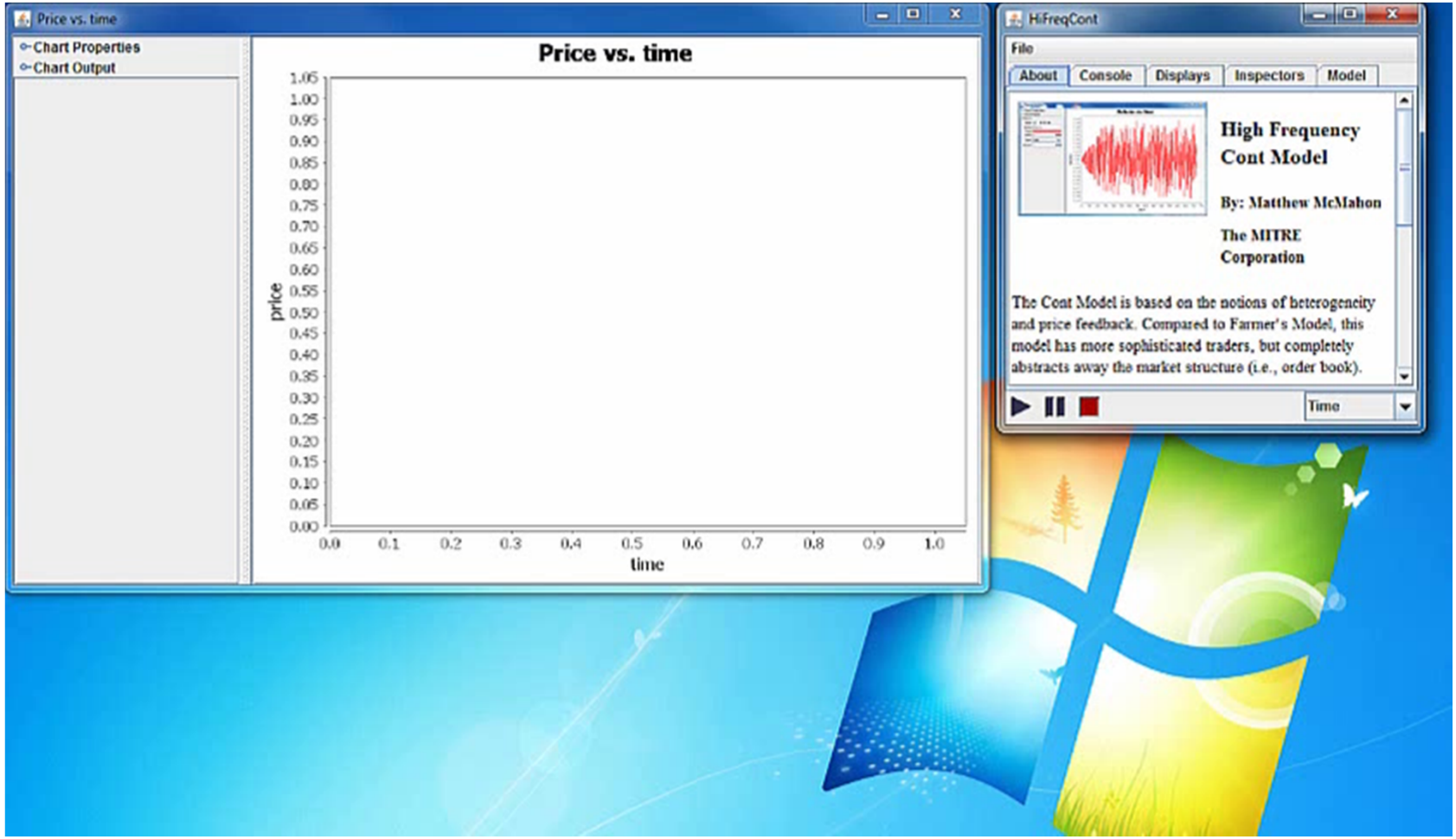
- **Cont's Heterogeneous Feedback Model**
  - Distinct input parameters
  - **Two** output files
  - Single Market Model
  - Trader behavior is based on a reaction to information
- **Script for statistical analysis of model output**
  - Written in R



Ghoulmie, Cont, and Nadal. 'Heterogeneity and feedback in an agent-based market model,' J. Phys.: Condens. Matter 17 (2005) S1259–S1268.

MITRE

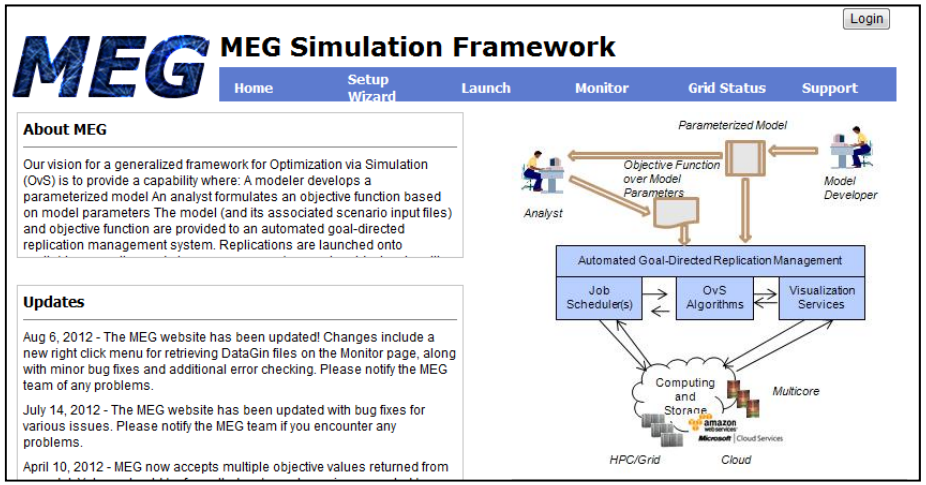
Case Study: Financial Market Model



MITRE

Case Study: Financial Market Model

- **MEG Implementation Steps**
  - Convert to headless
    - 7 parameters converted to command line input
  - Change file paths
  - Convert R script to Python
    - Run after each experiment
  - Create the Input Files
    - Configuration file
    - Condor submit file
    - Run script
  - Upload and Run
    - Upload the model and the script to the HIVE
    - Upload input files to the web interface



MITRE

Case Study: Financial Market Model

C:\Users\ceharvey\Documents\MEG\FSO\RunWithDG\runmodel.sh - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

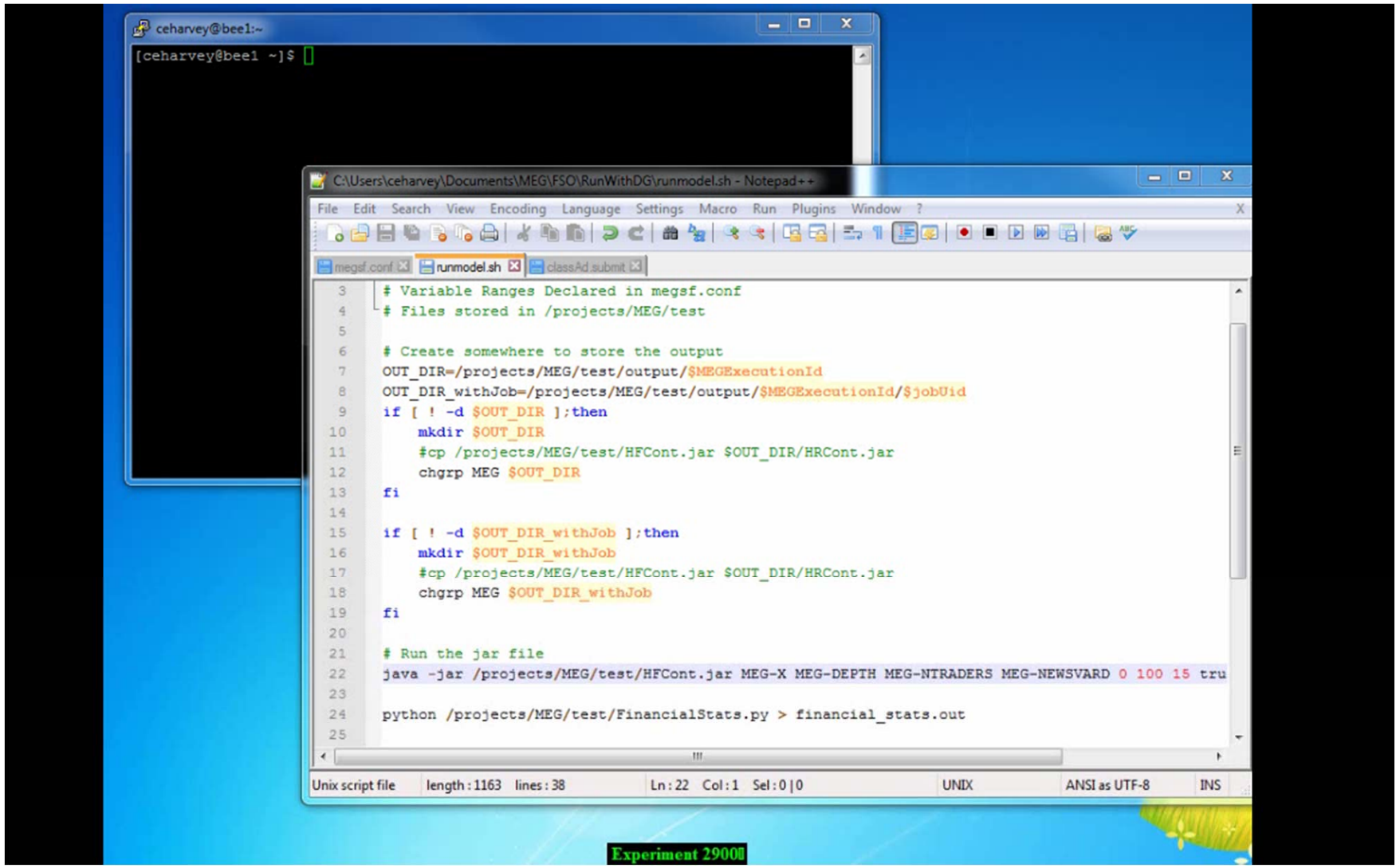
megsf.conf | runmodel.sh | classAd.submit

```
#!/bin/sh
# Run MEG for HF Cont Model
# Variable Ranges Declared in megsf.conf
# Files stored in /projects/MEG/test

# Create somewhere to store the output
OUT_DIR=/projects/MEG/test/output/$MEGExecutionId
OUT_DIR_withJob=/projects/MEG/test/output/$MEGExecutionId/$jobUid
if [ ! -d $OUT_DIR ];then
    mkdir $OUT_DIR
    #cp /projects/MEG/test/HFCont.jar $OUT_DIR/HRCont.jar
    chgrp MEG $OUT_DIR
fi

if [ ! -d $OUT_DIR_withJob ];then
    mkdir $OUT_DIR_withJob
    #cp /projects/MEG/test/HFCont.jar $OUT_DIR/HRCont.jar
    chgrp MEG $OUT_DIR_withJob
fi

# Run the jar file
java -jar /projects/MEG/test/HFCont.jar MEG-X MEG-DEPTH MEG-NTRADERS MEG-NEWSVARD 0 500 5 true > output.txt

python /projects/MEG/test/FinancialStats.py > financial_stats.out

# Echo inputs and outputs into a file to feed into DataGin if desired
#echo MEG-X MEG-Y MEG-Z $fSum $fProduct $MEGExecutionId $jobUid >output.$jobUid.txt
chgrp MEG /projects/MEG/test
cp param* $OUT_DIR_withJob/params.csv
cp price* $OUT_DIR_withJob/price_and_volume.csv
cp stats* $OUT_DIR_withJob/stats.csv
rm -rf *.csv
rm -rf output.txt
rm -rf financial_stats.out

##Return sum and product in multi-objective format
echo $jobUid

```

Unix script file | length : 1162 | lines : | Col : 4 | Sel : 0 | 0 | UNIX | ANSI as UTF-8 | INS

Experiment 29008

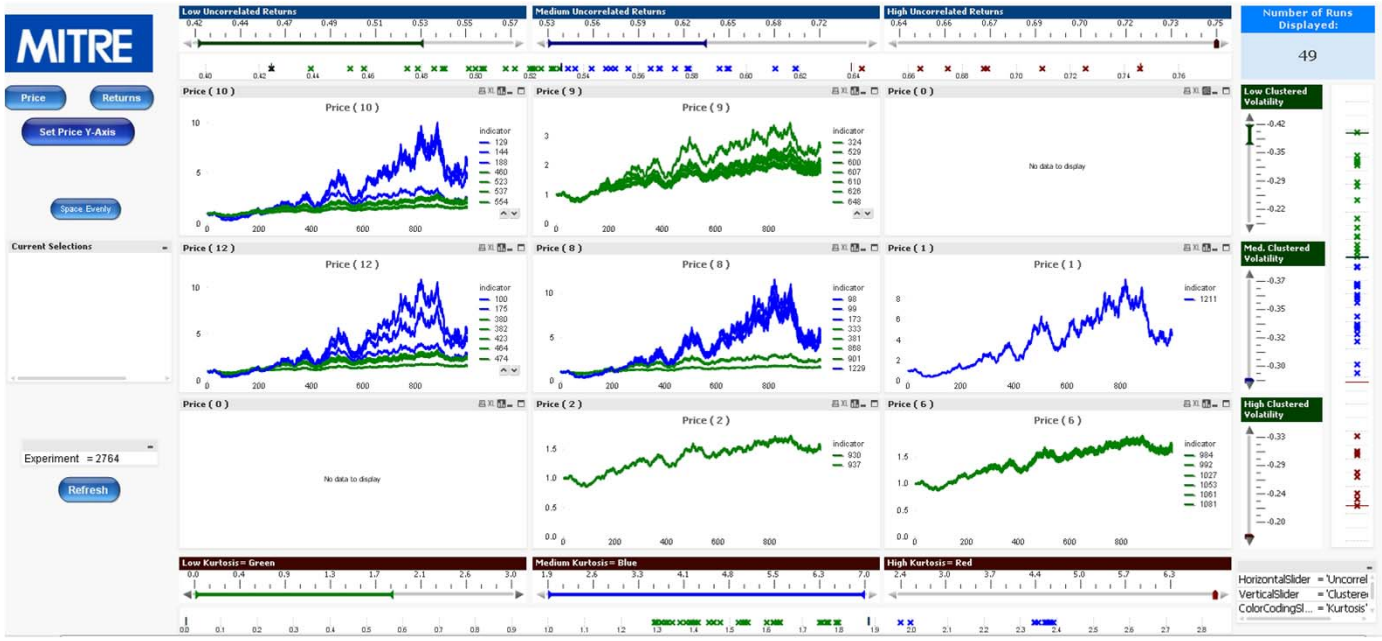
Case Study: Financial Market Model



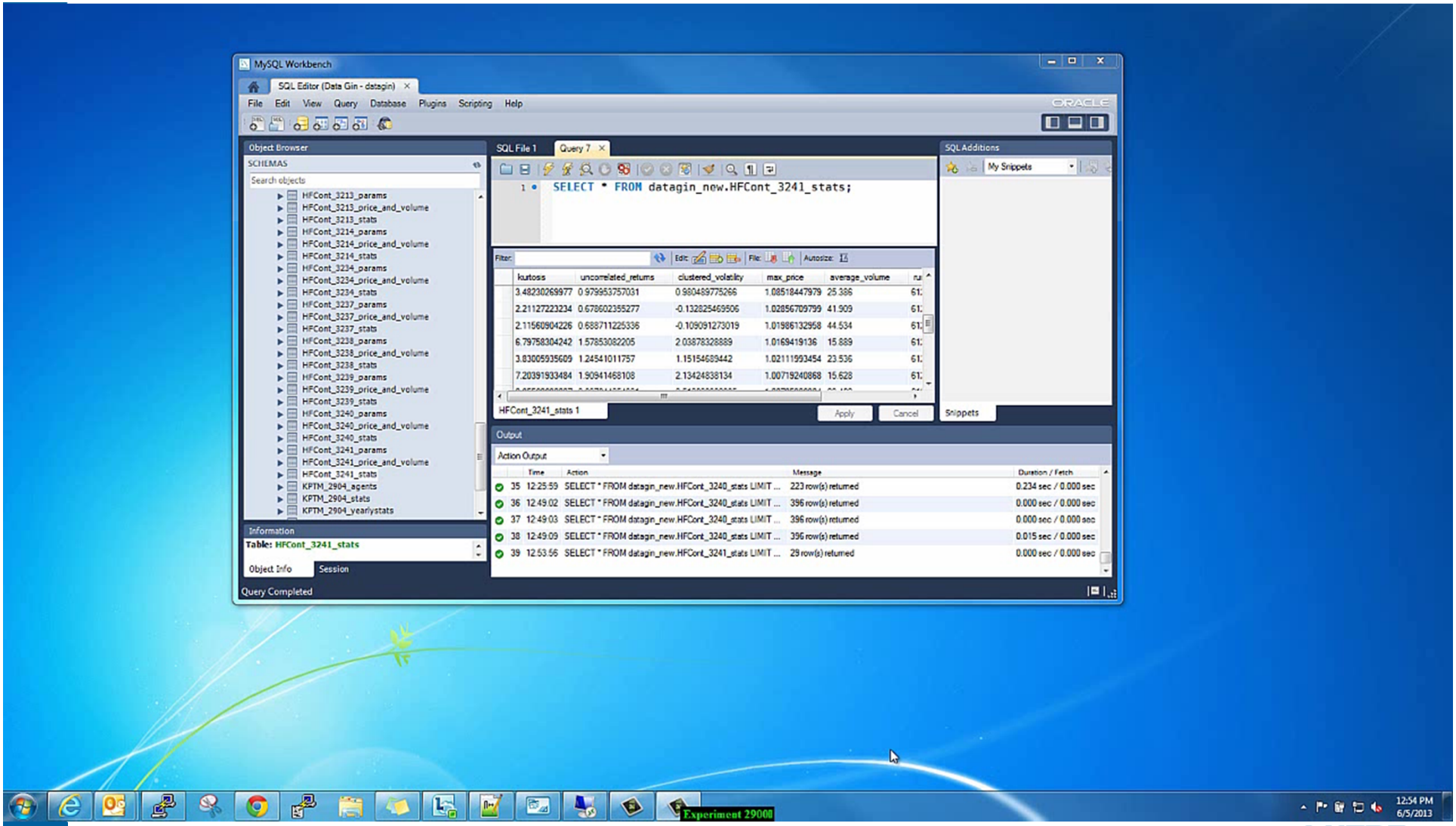
MITRE

Case Study: Financial Market Model

- **“Fast-Time” Visualization**
  - Qlikview connected to the Data Gin
  - Micro-Maps format
  - Interactive visualization tool that allows results to be seen as experiments are completed



Case Study: Financial Market Model



MITRE

Contact

Christine Harvey

The MITRE Corporation

ceharvey@mitre.org

MITRE