

Development of open source software for power market research: the AMES test bed

Hongyan Li

Electrical and Computer Engineering (ECpE) Department, Iowa State University, Ames, IA 50011-1070, USA; email: lihy@iastate.edu

Leigh Tesfatsion

Department of Economics, Iowa State University, Ames, IA 50011-1070, USA; email: tesfatsi@iastate.edu

Open source software (OSS) expresses the idea that developers should be able to license the publication of their software in a manner permitting anyone to freely use, modify and distribute the software. Today OSS is widely used in the software industry: for example, for language development tools (eg, NetBeans for Java), office document processors (eg, OpenOffice) and operating systems (eg, Linux, OpenSolaris). Yet OSS has been slow to penetrate the power industry; heavy reliance is still placed on closed-source commercial software packages. Open source software tends to be used for specialized purposes (eg, circuit design) rather than for the general-purpose analysis of power systems. This study discusses the potential benefits and drawbacks of developing OSS for power market research, using the AMES Wholesale Power Market Test Bed for concrete illustration.

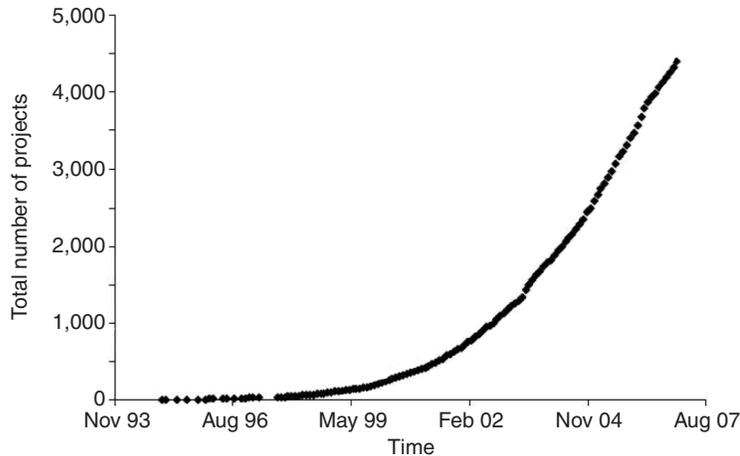
1 INTRODUCTION

Before the 1980s, being a programmer required highly specialized training. It was difficult for most consumers of commercial software to understand the architecture and specific content of the software they were using and hence to ensure it met all of their application requirements.

Richard Stallman, a pioneer of the free software movement, founded the GNU Project (“GNU’s Not Unix”) in September 1983 and the Free Software Foundation in October 1985. In 1998, a group of individuals meeting in Palo Alto, CA, proposed the term “open source software” (OSS) as a way of expressing the idea that developers should be able to publish their software under a license permitting anyone to freely use, modify and distribute it. Fogel (2009) provides a detailed history of these events.

The authors thank members of the ISU Electric Energy Economics Group (E3G) 2009 (www.econ.iastate.edu/tesfatsi/E3GroupISU.htm) for many useful discussions related to the topic of this study.

FIGURE 1 OSS project growth from November 1993 through August 2007.



Source: Deshpande and Riehle (2008).

The advantage of OSS is that people interested in the software can read the detailed code, understand the related design and modify the code to suit their requirements. People do not have to “reinvent the wheel” each time they need a specific code design for a specific problem. Instead, they can start from a previously established foundation subjected to open peer review, and they can do so without having to pay anything.

The exponential growth of OSS from 1993 through early 2008 is documented by Deshpande and Riehle (2008) (see Figure 1). Open source software is now widely used in the software industry: for example, for language development tools (eg, NetBeans for Java), office document processors (eg, OpenOffice) and operating systems (eg, Linux, OpenSolaris). Open source software is now increasingly written as commercial-grade software, which could represent a significant change in the traditional proprietary approach to software development.

Within the power industry, however, OSS development is still very much the exception rather than the rule. The question then arises: will OSS play an important role in future power industry application software?

This study discusses the potential benefits and drawbacks of OSS design and development for power market research. The AMES Wholesale Power Market Test Bed (Tsfatsion (2009a)) is used for concrete illustration. AMES is an open source, agent-based computational laboratory specifically designed for the systematic exploration of strategic trading in restructured wholesale power markets operating over AC transmission grids.

2 OVERVIEW

2.1 Why OSS for power market research?

Electric power systems are extremely complicated, comprised of generation, transmission and distribution subsystems. In order to keep modern power systems operating effectively and smoothly, different kinds of software are used at different operational stages and at different operational levels.

For example, supervisory control and data acquisition (SCADA) and energy management system (EMS) software is used to supervise, control and manage generation and transmission systems, and distribution management system (DMS) software is used to manage distribution networks. Also, relays installed at critical points along a power grid use embedded software to protect against sudden changes in current, voltage or frequency that could cause damage.

Traditionally, the power industry has relied on commercial software companies to provide the software that is needed for SCADA, EMS, DMS and other basic operational uses. This commercial software has typically been licensed in such a way that code is kept hidden, even to licensed users, as a protection against free-riding by rival vendors. If software is designed to solve well-understood problems, such as ones involving SCADA, then customers can test the accuracy and efficiency of the software against competing vendor products without delving into implementation details. In this case, hidden code is typically not a problem.

The recent restructuring of power markets, however, has led to new kinds of customer service requirements, such as two settlement systems, locational marginal pricing, ancillary service markets and advanced metering. These requirements continue to evolve.

Keeping code hidden in circumstances involving fundamentally new system design features, such as restructured power markets, can be counterproductive. Lack of open source access to code prevents customers from gaining a complete and accurate understanding of what has been implemented, restricts the ability of customers to experiment with new software features and hinders customers from tailoring software to specific evolving needs. Moreover, it prevents the evaluation of new system design features by objective outsiders, eg, university researchers.

Another issue that potentially favors the use of OSS for restructured power markets is the maintenance of code quality. At present, there is intense competition among rival vendors to supply software for new customer service requirements in these markets. However, once a market operator has committed to a vendor for a particular type of application software and the vendor customizes this software for particular energy region requirements, the market operator becomes “locked in” to the vendor. That is, the variable operational costs associated with the use of the current vendor’s customized software are relatively small, whereas the fixed operational costs associated with switching to a new vendor would be relatively large.

The vendor thus becomes a monopolistic supplier of customized software to its energy region customers. This reduces the vendor’s incentives to upgrade its customized software using the latest computer technology developments. It also puts market operators in a weak position with regard to any demands they might have for access to code, design features and implementation details.

On the other hand, in fairness to vendors, the development of software in order to meet the new service requirements of restructured power markets is costly in terms of time and money because it must be specialized to the particular circumstances of each energy region. It is understandable, then, that commercial software companies should want to develop this software as closed rather than open source in order to protect their investments from rival vendors.

In addition, incentives for keeping code hidden and for being cautious about upgrades also flow from the side of energy region customers. Market operators for restructured power markets could have legitimate security concerns regarding the release of code detailing the implementation of their market operations. Moreover, market operators would surely be concerned with potential disruptions of their operations should their software vendors upgrade features of their customized software in such a way that backward compatibility with legacy software is not preserved. The need for vendors of customized software to retain backward compatibility with legacy software products currently being used by their customers can result in large maintenance costs. Programmers thoroughly familiar with this legacy code must therefore be retained, or recruited and trained, even if the legacy code is essentially obsolete.

Open source software is based on distributed parallel development with open access to code, the antithesis of monopolistic vendor–customer relationships. The maintenance and upgrading of OSS application software is widely shared by distributed programmers, unconstrained by backward compatibility requirements stemming from purely financial concerns.

Important issues remain, however, regarding the benefits and drawbacks of OSS for power market application relative to closed-source commercial software. One key question concerns the degree to which successive version releases by distributed programmers can and should retain backward compatibility and a consistent design purpose. A second question concerns the degree to which distributed programmers can ensure code maintenance and support as well as continual code development. These two questions are addressed in the next section.

2.2 General design and development issues

2.2.1 *Backward compatibility and consistent design purpose*

Typically, the development process for any given software project proceeds in stages, through successive version releases, during which new software features and modules are developed and tested. Each successive version release is accompanied by

supporting material that highlights the project goals, summarizes the manner in which the current version release supports these goals, outlines the intended scope of application of the new release, clarifies the extent to which backward compatibility with previous version releases is retained and explains technical requirements for user implementation.

During this multi-stage development process, as technical details are studied and become increasingly well understood, it is sometimes necessary to adjust project goals to more realistic expectations. On the other hand, unforeseen advances in computer technology can enable enhancements and extensions of project objectives that were not originally considered to be practical.

If the development of a software project is taking place within the confines of a hierarchical organization such as a corporate firm, successive modifications of project goals and software design features can, in principle, be top-down coordinated so that an appropriate degree of consistency is retained from one version release to the next. This consistency tends to encourage cumulative software development, meaning that new version releases build on previous version releases and hence retain a core identity over time.

The situation for OSS development, however, is more problematic. The developers of OSS applications are, by definition, not required to function within the confines of a hierarchical organization. In particular, there are no commonly shared financial considerations forcing developers to retain backward compatibility with previous version releases.

This freedom from financial backward-compatibility concerns is certainly a plus to the extent that it encourages widespread open-ended experimentation with new software design features. On the other hand, substantial interface changes and other design features that require would-be users to undertake extensive retraining and legacy code modifications can be extremely disruptive and therefore discourage adoptions.

In response to these well-known problems, OSS developers have increasingly used the Internet to form voluntary communities centered around particular software development projects, with the help of umbrella sites such as Sourceforge.net.¹ They have also developed new tools (eg, Subversion²) that enable organized storage of and access to successive OSS version releases (see Wheeler (2007)).

Recent efforts along these lines have also been undertaken specifically for the power systems community. In June 2008, the Institute of Electrical and Electronics Engineers (IEEE) officially recognized the newly organized IEEE Taskforce on Open Source Software for Power Systems with an accompanying website for posting relevant resources. In addition, a special website has been developed, entitled “Open

¹ Sourceforge.net is a repository of open source software.

² Subversion is an open source version control system. It can be found at <http://subversion.tigris.org>.

source software for electricity market research, teaching, and training”, that caters more specifically to OSS power market applications.³

Members of the power systems community are being encouraged to develop their software applications as OSS that can be posted at these sites. The extent to which these and other efforts will help to achieve a balance between innovation and practical usefulness for open source software power systems that is workable in the long run remains to be seen.

2.2.2 Maintenance, support and continual development

The OSS development process can be compared to an assembly line in which different programmers fulfill different tasks using different tools to help ensure software quality and efficiency. As will be concretely illustrated in Section 3, OSS development tools have substantially improved over the past decade. However, the question remains: what will ensure the continual availability of programmers willing and able to maintain, support and develop OSS?

First-generation OSS programmers were idealists. They were interested in proving that practically useful software could be developed and distributed under open-access principles permitting adopters to freely read and modify the code. Many contemporary OSS developers retain the essential spirit of these first-generation OSS programmers. Starting from real-world requirements, they seek to add features that render their OSS faster, more functional and more reliable. Open source software changes projects are currently being developed at all levels, from large-scale commercial-grade applications to smaller-scale applications geared more toward research and teaching.

A common business model for commercial-grade OSS development is to provide the basic OSS for free while charging fees for software training, support and consulting. In addition, fees are sometimes charged for customized module development. For example, some power consulting companies have adopted this pricing strategy. To the extent that this business model is commercially successful, sustaining a demand for OSS programmers should not be a problem.

Moreover, the potential for sustaining a supply of OSS programmers also appears promising. Twenty years ago, few people were knowledgeable enough to engage in software development. Now, however, university graduate programs in many disciplines (eg, power engineering) routinely expect MSc and PhD students to acquire good computer programming skills.

Graduates of these university programs are potential developers of commercial OSS in a wide variety of application areas. Moreover, to the extent that the academic training of university graduates continues to stress the importance of open

³ Website hosted by the Economics Department of Iowa State University: www.econ.iastate.edu/tesfatsi/ElectricOSS.htm.

communication to both permit and encourage the accumulation of research findings, graduates who remain in academia should provide a pool of potential OSS developers for research and teaching purposes.

Given the time and effort needed for OSS development, grants will be needed to sustain these university OSS development efforts. No doubt government grants will remain an important source of funding for university researchers focused on basic OSS development issues (eg, improved integrated development environments). However, ongoing collaborative support from the industrial sector will also be essential to ensure that university OSS developers stay in tune with practical real-world needs.

For concrete illustration, the following section discusses an ongoing university project involving the development of AMES, an OSS test bed for the explorative study of restructured wholesale power markets. This project has been supported by the National Science Foundation and by the ISU Electric Power Research Center, a power industry consortium.

3 OSS POWER MARKET APPLICATION: THE AMES TEST BED

3.1 Purpose

In an April 2003 White Paper, the US Federal Energy Regulatory Commission proposed a new market design for US wholesale power markets (FERC 2003). Over half of the US generating capacity is now operating within the footprint of a wholesale power market restructured in compliance with the basic provisions of the FERC's design.

These restructured wholesale power markets are complex, involving physical constraints, complicated market protocols and behavioral dispositions of human participants. Moreover, time series are short due to the relative recency of restructuring efforts, and the data that is available is often only released with a delay and only in a partially masked form. Consequently, it is difficult to model and study these markets using standard analytical and statistical tools.

An additional complicating factor is that many economists are not familiar with transmission grid aspects of power systems, so they often focus on highly simplified two-bus or three-bus systems. Conversely, many power engineers are not familiar with basic economic market concepts, let alone the complicated design of restructured wholesale power markets. Modeling efforts by interdisciplinary teams capable of addressing both engineering and economic concerns would therefore be highly desirable.

In response to these concerns, an interdisciplinary group of researchers at Iowa State University has undertaken the OSS development of a wholesale power market test bed (referred to as AMES, for agent-based modeling of electricity systems). The AMES test bed permits the systematic experimental study of strategic trading

behaviors within restructured wholesale power markets operating over realistically rendered AC transmission grids. In addition, AMES facilitates the augmentation of empirical input data with simulated input data to permit the study of a broader array of scenarios.

From the beginning, AMES was designed for research and teaching purposes rather than for commercial-grade application. AMES is entirely developed in the widely used Java programming language in order to facilitate readability and use. AMES is entirely OSS, combining a collection of basic OSS modules for learning representation, optimal power flow solution, graphic display and other functions. The modular and extensible OSS architecture of AMES permits users to modify and extend the code with relative ease to suit their specific requirements.

The first version of AMES was released as OSS at the IEEE Power and Energy Society General Meeting (PES GM) in 2007, and a substantially expanded second version was released as OSS at the IEEE PES GM in 2008. Downloads, manuals and tutorial information for all AMES version releases to date are accessible at the AMES homepage (Tesfatsion (2009a)). AMES is also available for download at the software site of the IEEE Task Force on Open Source Software for Power Systems.⁴

The release of AMES as OSS is intended to encourage the cumulative development of this test bed by multiple researchers in directions appropriate for their specific needs. It is also intended to encourage continuing dialog with market stakeholders and regulators leading to successive refinements and improvements of the test bed.

3.2 Key features

The latest version of AMES (v2.02) incorporates, in simplified form, core features of the wholesale power market design proposed by the US FERC (FERC 2003) (see Figure 2 on the facing page). A detailed description of many of these features can be found in Sun and Tesfatsion (2007) and Li *et al* (2009). Below is a summary description of the logical flow of events in the AMES wholesale power market as currently implemented.

The AMES wholesale power market operates over an AC transmission grid starting on day 1 and continuing through a user-specified maximum day (unless terminated earlier in accordance with a user-specified stopping rule). Each day D consists of 24 successive hours, $H = 00, 01, \dots, 23$.

The AMES wholesale power market includes an independent system operator (ISO) and a collection of energy traders consisting of load-serving entities (LSEs) and generation companies (GenCos) distributed across the buses of the transmission grid. Each of these entities is implemented as a software program encapsulating both methods and data; see, for example, the schematic depiction of a GenCo in Figure 3 on the facing page.

⁴ See http://ewh.ieee.org/cmte/psace/CAMS_taskforce/index.htm.

FIGURE 2 AMES test bed architecture.

-
- > **Traders**
 - LSEs (bulk-power buyers)
 - GenCos (bulk-power sellers with learning capabilities)
 - > **Independent system operator (ISO)**
 - Day-ahead hourly scheduling via bid/offer-based DC-OPF
 - System reliability assessments
 - > **Two-settlement process**
 - Day-ahead market (double auction, financial contracts)
 - Real-time market (settlement of differences)
 - > **AC transmission grid**
 - LSEs and GenCos located at user-specified buses across the transmission grid
 - Congestion managed via locational marginal pricing
-

FIGURE 3 AMES GenCo: a cognitive agent with learning capabilities.

```

Public Access:
// Public Methods
getWorldEventSchedule (clock time);
getMarketProtocols (supply offer reporting, settlement, ...);
getMarketProtocols (ISO market power mitigation);
Methods for receiving data;
Methods for retrieving stored generator data.

Private Access:
// Private Methods
Methods for gathering, storing, and sending data;
Methods for calculating my expected/actual net earnings;
Method for updating my supply offers (LEARNING).
// Private Data
My grid location, cost function, capacity, current wealth, ...;
Historical data (cleared supply offers, LMPS, ...);
Address book (communication links).
    
```

The objective of the ISO is the reliable attainment of appropriately constrained operational efficiency for the wholesale power market, ie, the maximization of total net benefits subject to generation and transmission constraints.

In an attempt to attain this objective, the ISO undertakes the daily operation of a day-ahead market settled by means of locational marginal pricing (LMP), ie, the determination of prices for electric power in accordance with both the location and timing of its injection into, or withdrawal from, the transmission grid.

The objective of each LSE is to secure power for its downstream (retail) customers. During the morning of each day D , each LSE reports a demand bid to the ISO for

the day-ahead market for day $D + 1$. Each demand bid consists of two parts: a fixed demand bid (ie, a 24-hour load profile) and 24 price-sensitive demand bids (one for each hour), each consisting of a linear demand function defined over a purchase capacity interval. LSEs have no learning capabilities; LSE demand bids are user specified at the beginning of each simulation run.

The objective of each GenCo is to secure for itself the highest possible net earnings each day. During the morning of each day D , each GenCo i uses its current action choice probabilities to choose a supply offer from its action domain AD_i to report to the ISO for use in all 24 hours of the day-ahead market for day $D + 1$. Each supply offer in AD_i consists of a linear marginal cost function defined over an operating capacity interval. GenCo i 's ability to vary its choice of a supply offer from AD_i permits it to adjust the ordinate/slope of its reported marginal cost function and/or the upper limit of its reported operating capacity interval in an attempt to increase its daily net earnings.

After receiving demand bids from LSEs and supply offers from GenCos during the morning of day D , the ISO determines and publicly reports hourly power supply commitments and LMPs for the day-ahead market for day $D + 1$ as the solution to hourly bid/offer-based DC optimal power flow (DC-OPF) problems. Transmission grid congestion is managed by the inclusion of congestion cost components in LMPs.

At the end of each day D , the ISO settles all of the commitments for the day-ahead market for day $D + 1$ on the basis of the LMPs for the day-ahead market for day $D + 1$.

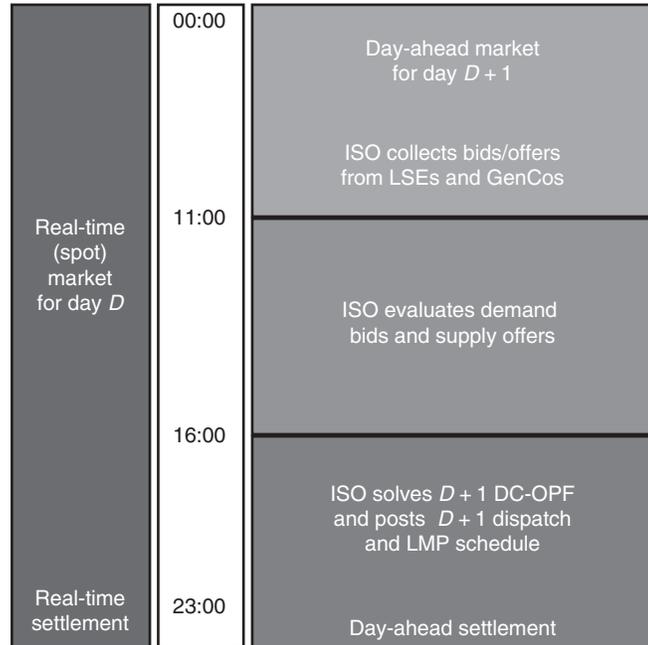
At the end of each day D , each GenCo i uses stochastic reinforcement learning to update the action choice probabilities currently assigned to the supply offers in its action domain AD_i , taking into account its day D settlement payment (“reward”). If the supply offer reported by GenCo i on day D results in a relatively good (respectively poor) reward, GenCo i increases (respectively decreases) the probability of choosing this supply offer on day $D + 1$.

There are no system disturbances (eg, weather changes) or shocks (eg, forced generation outages or line outages). Consequently, the binding financial contracts determined in the day-ahead market are carried out as planned and traders have no need to engage in real-time (spot) market trading.

Each LSE and GenCo has an initial holding of money that changes over time as it accumulates earnings and losses. There is no entry of traders into, or exit of traders from, the wholesale power market. LSEs and GenCos are currently allowed to go into debt (negative money holdings) without penalty or forced exit.

The flow of activities for the ISO on a typical day D is depicted in Figure 4 on the facing page. Data flow for the principal types of AMES traders participating in the day-ahead market is depicted in Figure 5 on page 122. Several of the basic AMES module components that implement this flow of activities are schematically

FIGURE 4 AMES ISO activities during a typical day D .



depicted in Figure 6 on page 123 for the special case of a five-bus transmission grid with no system shocks or disturbances.

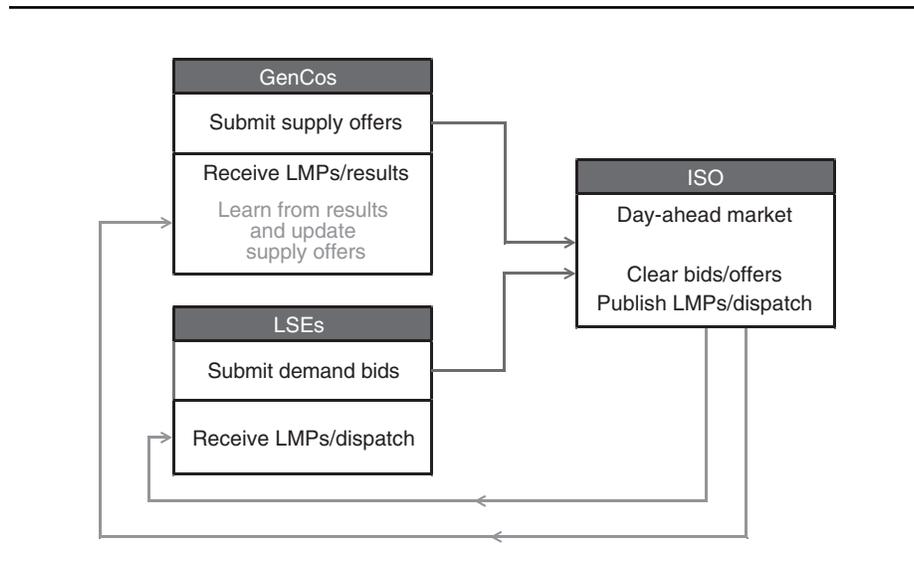
As seen in Figure 7 on page 124, AMES has a graphical user interface (GUI) with separate screens for carrying out the following functions:

- 1) creation, modification, analysis and storage of case studies;
- 2) initialization and editing of the structural attributes of the transmission grid;
- 3) initialization and editing of the structural attributes of LSEs and GenCos;
- 4) specification of learning parameters for GenCos;
- 5) specification of simulation controls (eg, the simulation stopping rule); and
- 6) customization of table and chart output displays.

3.3 Development tools used

As seen in Figure 8 on page 124, various OSS tools were used in the development of AMES (v2.02). A more careful description of these tools is given below.

FIGURE 5 Data flow for AMES traders in the day-ahead market.



Java development kit (JDK): version 6, update 1 (6u1) of the Java SE Development Kit⁵ was used to develop the basic AMES code.

Java integrated development environment (IDE): the NetBeans IDE 6.0⁶ was used in conjunction with the Java development kit to develop AMES as a standard Java package. The NetBeans IDE is a powerful open source cross-platform tool for Java programming.

Java chart library: JFreeChart⁷ is an OSS chart library that is used to create different kinds of charts in AMES.

Java agent-based toolkit: the OSS agent-based toolkit RepastJ⁸ is used to control agent activities in AMES.

Java DC-OPF solver (DCOPFJ): DCOPFJ, a free open source solver for DC optimal power-flow problems developed by Sun and Tesfatsion (see Tesfatsion (2007b)), is used by the AMES ISO to solve hourly bid/offer-based DC-OPF problems.

Java reinforcement learning module (JReLM): JReLM, an open source Java learning module developed by Gieseler (2005), is used to control agent learning behaviors in AMES.

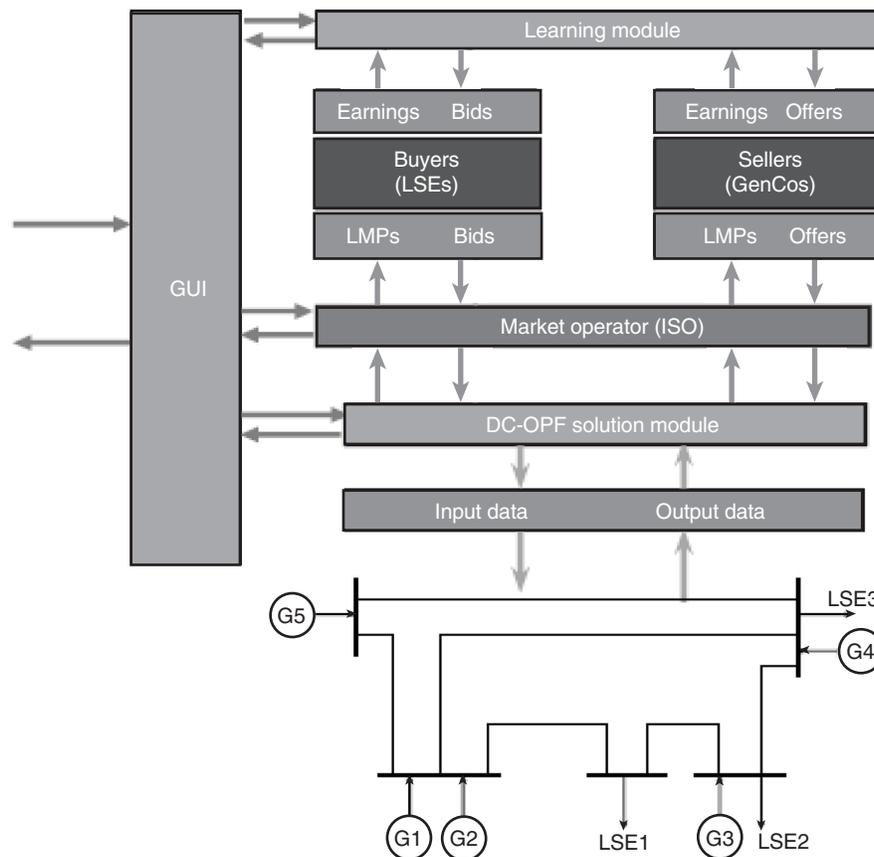
⁵ Available from <http://java.sun.com>.

⁶ Available from www.netbeans.org.

⁷ Available from www.jfree.org/jfreechart.

⁸ Version 3.1 available from http://repast.sourceforge.net/repast_3/download.html.

FIGURE 6 Illustrative depiction of basic AMES module components: learning (JReLM); DC-OPF solver (DCOPFJ); transmission grid; and graphical user interface (GUI).



Colt libraries for Java high-performance computing: Colt provides a set of open source libraries for high-performance scientific and technical computing in Java.⁹

Subversion for version management: Subversion is an open source version control system facilitating simultaneous software development by multiple project members.

3.4 Licensing and release

The AMES Wholesale Power Market Test Bed is licensed by the copyright holders (Hongyan Li, Junjie Sun and Leigh Tesfatsion) as free open source software under

⁹ The Colt Project can be found at <http://acs.lbl.gov/~hoschek/colt/>.

FIGURE 7 AMES GUI.

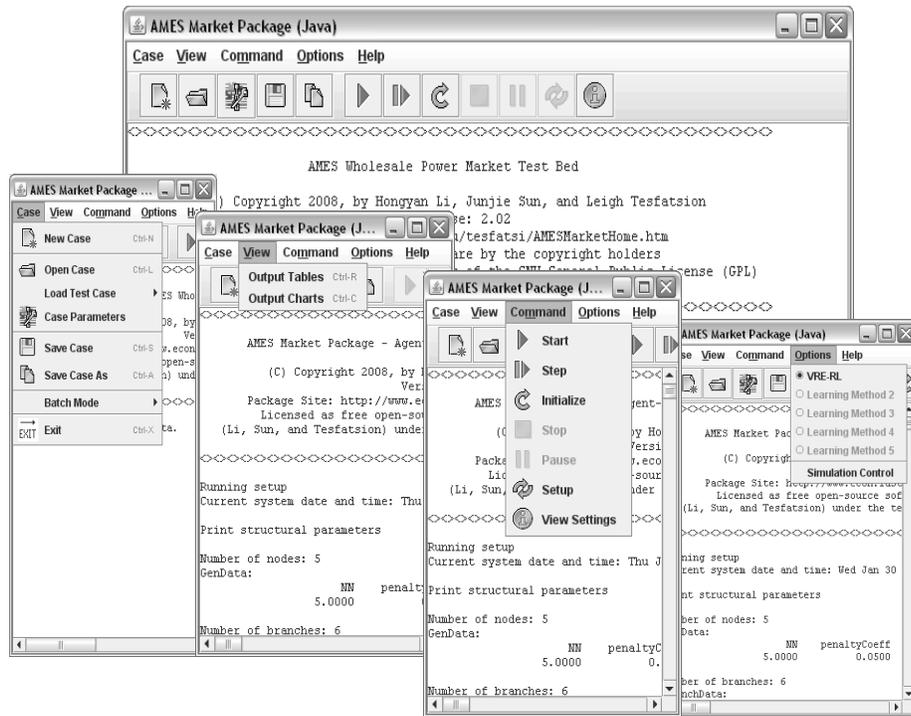
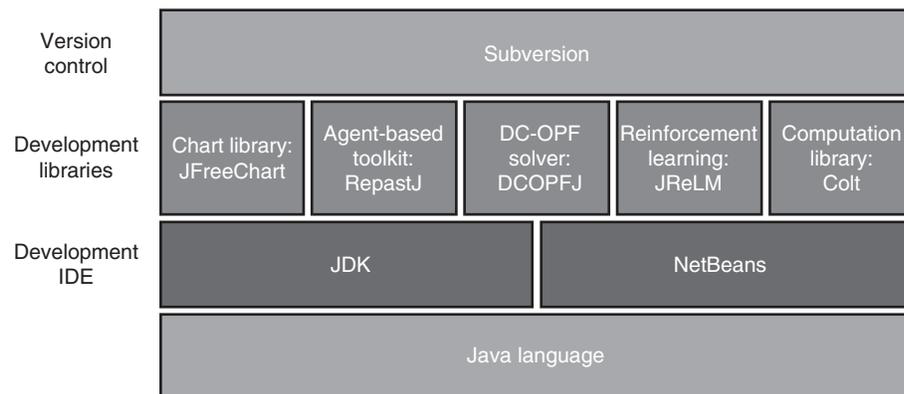


FIGURE 8 OSS tools used in the development of AMES (v2.02).



the terms of the GNU General Public License.¹⁰ Anyone who is interested is allowed to view, modify and/or improve upon the code for AMES, but any software generated using all or part of this code must be released as free open source software in turn. The latest version of AMES can be downloaded in a zip file from the AMES homepage (Tsfatsion (2009a)). This zip file includes three help files and three file directories, as follows:

AMESMarketReadMe.htm: this help file is a slightly modified version of the AMES homepage. It provides useful information about AMES, including an overview of capabilities, pointers to current and previous versions of AMES for download and annotated pointers to project papers and publications making use of AMES.

AMESMarketProjectSetupInfo.pdf: this help file is a basic manual for AMES. Included are instructions for setting up AMES as a Java project, loading and viewing AMES test cases, developing new AMES test cases, modifying the AMES source code and running AMES experiments in either individual or batch mode.

AMESVersionReleaseHistory.htm: this help file provides annotated pointers to all released AMES versions.

src directory: this directory includes all of the AMES source code.

lib directory: this directory includes all special OSS libraries used to implement AMES, eg, an executable Java archive (“jar”) file *Colt.jar* for *Colt*.

DATA directory: this directory provides sample data input files for five-bus and thirty-bus test cases that can be used as templates for the development of new AMES test cases. It also provides a sample batch-mode file that can be used as a template for setting up AMES experiments so that multiple runs can be implemented automatically in one batch.

3.5 Applications to date

The AMES test bed has been used to conduct systematic experimental studies focusing on various performance aspects of restructured wholesale power markets operating over AC transmission grids when profit-seeking GenCos have learning capabilities allowing them to strategically evolve their supply offers over time (Sun and Tsfatsion (2007); Li *et al* (2009)).

For example, AMES has been used to investigate how demand–bid price sensitivity affects dynamic wholesale power market performance, with a particular stress on LMP response (see Li *et al* (2008)). Results show that, even with 100% price-sensitive demand bids, average LMP is much higher under GenCo learning. An

¹⁰ See www.gnu.org/licenses/licenses.html.

implication of these findings is that active demand-side bidding by LSEs reflecting better integration of wholesale and retail markets is needed for the proper functioning of restructured wholesale power markets. Having countervailing power (ie, strategically selected demand bids by LSEs as well as strategically selected supply offers by GenCos) should result in more competitive pricing.

AMES has also been used to investigate how the imposition of a price cap on supply offers affects dynamic wholesale power market performance (see Li *et al* (2008)). Results show that a supply-offer price cap does not necessarily provide an upper bound on LMPs. Moreover, setting an inappropriately low level for the supply-offer price cap can cause other problems, such as increased price volatility and inadequacy (insufficient total generation capacity to meet a fixed load).

In addition, AMES has been used to investigate strategic capacity withholding by GenCos in restructured wholesale power markets under varied demand conditions (see Li and Tesfatsion (2008)). This strategic capacity withholding can involve both *physical* withholding (reporting of lower-than-true maximum operating capacity) and *economic* withholding (reporting of higher-than-true marginal costs). The ability of demand conditions to mitigate incentives for capacity withholding is systematically explored by letting demand bids incrementally adjust from 100% fixed demand to 100% price-sensitive demand.

The AMES software package has been downloaded from the AMES homepage by numerous researchers worldwide, and several research groups have indicated through communications with AMES team members that the AMES OSS has helped them in the design of their own project software. For example, research groups in Germany and Australia have used aspects of the AMES OSS to build agent-based test beds for the study of CO₂ emission control systems for power markets, and a research group in China has developed an AC OPF module for possible inclusion in a future version of AMES.

3.6 Future prospects

It is hoped that the free open source release of AMES (v2.02) will encourage the cumulative development of future versions with enhanced features critical for determining the performance of real-world restructured electricity markets. Examples of such enhanced features include the following:

- System shocks and/or disturbances leading to discrepancies between day $D-1$ financial contracts and day D required transactions (active two-settlement system).
- Enhanced transmission grid features.
- Incorporation of an AC OPF solver to permit DC versus AC OPF error comparisons.

Development of open source software for power market research: the AMES test bed 127

- Enhanced modeling of ISO-managed unit commitment taking into account start-up costs, down-time constraints and ramping constraints.
- Security constraints incorporated into DC/AC OPF problem formulations as a hedge against system disturbances.
- ISO-managed resource adequacy assessment.
- Emission constraints and other mandated environmental protection measures.
- Upstream fuel markets permitting more empirically based derivations of cost functions for GenCos.
- Incorporation of learning capabilities for LSEs.
- Additional types of learning methods for potential use by GenCos and LSEs (eg, anticipatory learning).
- Tracing and graphical display of GenCo learning characteristics over time (eg, action propensity and probability values) to permit a more detailed micro-level understanding of GenCo co-learning.
- Inclusion of bankruptcy rules to handle situations in which one or more traders use up all of their liquid assets.
- A financial transmission rights market to permit hedging against transmission congestion costs.
- Bilateral trading to permit longer-term contracting.
- Downstream retail markets permitting more empirically based derivations of LSE demand bids.

REFERENCES

- Deshpande, A., and Riehle, D. (2008). The total growth of open source. In *Proceedings of the Fourth Conference on Open Source Systems (OSS 2008)*. Springer.
- FERC (2003). Notice of White Paper, US Federal Energy Regulatory Commission.
- Fogel, K. (2009). *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly Media, Sebastopol, CA. (Also available through a Creative Commons Attribution–ShareAlike (3.0) License at <http://producingoss.com>.)
- Gieseler, C. (2005). A Java reinforcement learning module for the Repast toolkit: facilitating study and experimentation with reinforcement learning in social science multi-agent simulations. MS Thesis, Department of Computer Science, Iowa State University.
- Li, H., and Tesfatsion, L. (2009). Capacity withholding in restructured wholesale power markets: an agent-based test bed study. In *Proceedings of the IEEE Power Systems Conference and Exposition, Seattle, WA*. Version available at www.econ.iastate.edu/tesfatsi/CapacityWithholding.PSCE2009.LiTsfatsion.pdf.

- Li, H., Sun, J., and Tesfatsion, L. (2008). Dynamic LMP reponse under alternative price-cap and price-sensitive demand scenarios. In *Proceedings of the IEEE Power Engineering Society General Meeting, Pittsburgh, PA*. Version available at www.econ.iastate.edu/tesfatsi/DynamicLMPResponse.IEEEPEES2008.LST.pdf.
- Li, H., Sun, J., and Tesfatsion, L. (2009). Separation and volatility of locational marginal prices in restructured wholesale power markets. Working Paper, Economics Department, Iowa State University (in preparation).
- Somani, A., and Tesfatsion, L. (2008). An agent-based test bed study of wholesale power market performance measures. *IEEE Computational Intelligence Magazine* **3**(4), 56–72.
- Sun, J., and Tesfatsion, L. (2007a). Dynamic testing of wholesale power market designs: an open-source agent-based framework. *Computational Economics* **30**, 291–327.
- Sun J., and Tesfatsion, L. (2007b). DC optimal power flow formulation and solution using QuadProgJ. In *Proceedings of the IEEE Power and Energy Society General Meeting, Tampa, FL*. Version available at www.econ.iastate.edu/tesfatsi/DC-OPF.JSLT.pdf.
- Tesfatsion, L. (2009a). AMES wholesale power market test bed homepage. Website hosted by the Economics Department of Iowa State University: www.econ.iastate.edu/tesfatsi/AMESMarketHome.htm.
- Tesfatsion, L. (2009b). DCOPFJ: a DC optimal power flow solver (in Java). Website hosted by the Economics Department of Iowa State University: www.econ.iastate.edu/tesfatsi/DCOPFJHome.htm.
- Wheeler, D. A. (2007). Why open source software/free software? Look at the numbers! Available at www.dwheeler.com/oss_fs_why.html.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.