RVA: 3-D Visualization and Analysis Software to Support Management of Oil and Gas Resources

Final Scientific/Technical Report

Project Start Date: October 1, 2010

Project End Date: September 15, 2015

Donald A. Keefer, Eric G. Shaffer, Brynne Storsved, Mark Vanmoer, Lawrence Angrave, James R. Damico, Nathan Grigsby

December 2015

DOE Award Number: DE-FE0005961

Submitting Organization:
Illinois State Geological Survey,
University of Illinois at Urbana-Champaign
615 E. Peabody Drive
Champaign, IL 61820

Abstract

A free software application, RVA, has been developed as a plugin to the US DOE-funded ParaView visualization package, to provide support in the visualization and analysis of complex reservoirs being managed using multi-fluid EOR techniques. RVA, for Reservoir Visualization and Analysis, was developed as an open-source plugin to the 64 bit Windows version of ParaView 3.14. RVA was developed at the University of Illinois at Urbana-Champaign, with contributions from the Illinois State Geological Survey, Department of Computer Science and National Center for Supercomputing Applications. RVA was designed to utilize and enhance the state-of-the-art visualization capabilities within ParaView, readily allowing joint visualization of geologic framework and reservoir fluid simulation model results. Particular emphasis was placed on enabling visualization and analysis of simulation results highlighting multiple fluid phases, multiple properties for each fluid phase (including flow lines), multiple geologic models and multiple time steps. Additional advanced functionality was provided through the development of custom code to implement data mining capabilities. The built-in functionality of ParaView provides the capacity to process and visualize data sets ranging from small models on local desktop systems to extremely large models created and stored on remote supercomputers. The RVA plugin that we developed and the associated User Manual provide improved functionality through new software tools, and instruction in the use of ParaView-RVA, targeted to petroleum engineers and geologists in industry and research. The RVA web site (http://rva.cs.illinois.edu) provides an overview of functions, and the development web site (https://github.com/shaffer1/RVA) provides ready access to the source code, compiled binaries, user manual, and a suite of demonstration data sets. Key functionality has been included to support a range of reservoirs visualization and analysis needs, including: sophisticated connectivity analysis, cross sections through simulation results between selected wells, simplified volumetric calculations, global vertical exaggeration adjustments, ingestion of UTChem simulation results, ingestion of Isatis geostatistical framework models, interrogation of joint geologic and reservoir modeling results, joint visualization and analysis of well history files, location-targeted visualization, advanced correlation analysis, visualization of flow paths, and creation of static images and animations highlighting targeted reservoir features.

Table of Contents

Executive Summary

A free software application, RVA, has been developed on top of the US DOE-funded ParaView visualization package, to provide support in the visualization and analysis of complex reservoirs being managed using multi-fluid EOR techniques. RVA, for Reservoir Visualization and Analysis, was developed as an open-source plugin to the 64 bit Windows version of ParaView 3.14. RVA was developed at the University of Illinois at Urbana-Champaign, with contributions from the Illinois State Geological Survey, Department of Computer Science and National Center for Supercomputing Applications. RVA was designed to utilize and enhance the state-of-the-art visualization capabilities within ParaView, readily allowing joint visualization of geologic framework and reservoir fluid simulation model results. Particular emphasis was placed on enabling visualization and analysis of simulation results highlighting multiple fluid phases, multiple properties for each fluid phase (including flow lines), multiple geologic models and multiple time steps. Additional advanced functionality was provided through the development of custom code to implement data mining capabilities.

ParaView-RVA provides capabilities for easily visualizing multiple properties of the reservoir architecture, together with fluid variables like oil saturation, water pressure, and surfactant concentration, and even visualize fluid flow paths modeled from transient reservoir simulation. Attention was given to the User Manual to provide an easy-to-read document that provides not only instruction on using the customized tools in the RVA plugin, but in basic ParaView use as well. The built-in functionality of ParaView provides the capacity to process and visualize data sets ranging from small models on local desktop systems to extremely large models created and stored on remote supercomputers. The RVA plugin that we developed and the associated User Manual provide improved functionality through new software tools, and instruction in the use of ParaView-RVA, targeted to petroleum engineers and geologists in industry and research. The RVA web site (http://rva.cs.illinois.edu) provides an overview of functions, and the development web site (https://github.com/shaffer1/RVA) provides ready access to the source code, compiled binaries, user manual, and a suite of demonstration data sets. This simplifies user-community support of RVA through an industry standard community code development platform. The demonstration data sets also allow users the opportunity to explore the functionality of ParaView-RVA without having their own data sets in hand.

Key functionality has been included to support a range of reservoirs visualization and analysis needs, including:

- sophisticated connectivity analysis,
- cross sections through simulation results between selected wells,
- simplified volumetric calculations,
- global vertical exaggeration adjustments,
- ingestion of UTChem simulation results,
- ingestion of Isatis geostatistical framework models,
- interrogation of joint geologic and reservoir modeling results,
- joint visualization and analysis of well history files,

- location-targeted visualization,
- advanced correlation analysis,
- visualization of flow paths, and
- creation of static images and animations highlighting targeted reservoir features.

Through our efforts on this project, we found that using ParaView requires a bit more effort and expertise than is suggested within the ParaView web sites and documentation. Programming experience in Python or C++ also provides an advantage in simplifying complex analysis within ParaView. We also found that the code base of ParaView was sufficiently complex that extending ParaView with new code, as was done for this project, is much more effective if the programmers have prior experience programming within ParaView. With programmers who have sufficient experience, ParaView can be a productive and beneficial software package to work with, and development of custom tools can provide significant assistance to targeted user groups.

In discussions with geologists and petroleum engineers in small- to mid-sized upstream petroleum companies, we found a reluctance to explore ParaView-RVA. In contrast to this cool reception from petroleum company representatives, however, research groups in Universities and state geological surveys were generally very receptive and interested in exploring ParaView-RVA. These varied responses could be due to several factors. The development schedule established in our PMP, resulted in a product with limited functionality until late in the project life. This limited the utility of demonstrations or presentations at professional conferences or other venues. Future software development projects of this nature should consider setting up work plans so demonstrations and roll-out events are scheduled for after significant features have been created and tested. It appeared that a more focused marketing effort was needed to attract the attention of industry professionals, while research teams who had ready access to students and an emphasis on innovative research were more open to the software following brief discussions. In future projects, in addition to participating at national conferences, time and resources should be placed on creating brochures and demonstration videos that highlight functionality of interest to target groups; these can be made available both in-person and via project web sites. Consideration should also be given to obtaining exhibitor space at national conferences to provide hands-on demonstration opportunities with target user groups. Future software development projects could take advantage of this more directly by enlisting Universities into feature testing and advisory committee responsibilities.

In retrospect, the timing and number of the Go-No Go Decision Points in this project created logistical problems for efficient completion of the project goals. In development of our PMP and with the direction of our project manager, we scheduled 5 Go/No Go Decision Points to be in sync with incremental releases of the RVA software. While this seemed appropriate at the time, it turned out to be problematic. With our ambitious set of objectives, the final schedule did not allow sufficient time for feature testing and robust bug identification prior to each version release and Go/No Go decision point review. While we were able to meet our deadlines within this schedule, it resulted in a number of bugs that were not identified until after version releases, and that needed to be corrected during the no-cost extension periods. In addition to these logistical problems, our prior inexperience with Go/No Go decision points resulted in a misunderstanding of the project workflow during the decision-point review period. As a non-profit agency, the University does not have surplus revenue that could be charged if an unfavorable review led to a project discontinuation. Accordingly, University officers instructed us to discontinue working on the project during each Go/No Go review period. This resulted in 5 work stoppages and necessitated movement of contract-funded staff to other projects while the decision point was reviewed. Future software development projects would be better structured by reducing the number of Go/No Go decision points, and setting them significantly after software releases, thereby allowing sufficient time for feature testing and bug identification, and creating fewer disruptions of workflow and progress on objectives.

Further extension of ParaView-RVA offers significant promise and could be designed to provide much more support to energy-sector geoscience and engineering users. Additional functionality could include development of file readers for other popular geologic framework and reservoir simulation packages, development of new open file formats to better accommodate volumetric, transient simulations, to better interact with a range of GIS file types, and to better integrate functionality around wells and well data. Additionally, improved instruction, via the User Manual, on the use of the built-in ParaView Python calculator, with demonstrated examples of common problems, could provide important advice to improve the ease of use for new users.

Project Team and Advisory Committee

Through the length of the study, project staff members changed due to shifting needs and other obligations. Over the entire project, the senior RVA project staff consisted of:

Project Director/Principal Investigator: Mr. D. Keefer, ISGS
Co-Principal Investigator: Dr. E. Shaffer, CS
Co-Principal Investigator: Mr. J. Damico, ISGS
Programming lead (years 1-2): Dr. L. Angrave, CS
Programming lead (years 4-5): Mr. M. Vonmoer, NCSA
Technical assistant (years 2-5): Ms. B. Storsved, ISGS
Project geologists (years 4-5): Mr. N. Grigsby, ISGS and (year 4) Mr. R. Rice, ISGS.
Several student programmers were involved during the first 2-3 years of the project.

The Project Advisory Committee was comprised of:
Dr. M. Delshad, University of Texas-Austin
Dr. S. Frailey, University of Illinois at Urbana-Champaign
Mr. J. Grube, University of Illinois at Urbana-Champaign
Dr. H. Leetaru, University of Illinois at Urbana-Champaign
Dr. Y.-F. Lin, University of Illinois at Urbana-Champaign
Dr. E. Mehnert, University of Illinois at Urbana-Champaign
Dr. R. Okwen, University of Illinois at Urbana-Champaign
Dr. R. Ritzi, Wright State University
Mr. S. Whitaker, Devon Energy

The PAC members provided support to the project through participation in advisory meetings and through reviews of beta releases.

Project Objectives

This project was designed to produce an open-source software application, RVA, that would provide advanced 3-D visualization and analysis capabilities of data and simulation results to support management of unconventional oil and gas reservoirs. RVA was designed to be a plugin to the DOE-supported, open-source application, ParaView (http://www.paraview.org). RVA was designed to utilize and enhance the state-of-the-art visualization capabilities within ParaView, facilitating joint visualization of geologic framework and reservoir fluid simulation model results. Particular emphasis was placed on enabling visualization and analysis of simulation results highlighting multiple fluid phases, multiple properties for each fluid phase (including flow lines), multiple geologic models and multiple time steps. Additional advanced functionality was planned through the development of custom code to implement data mining capabilities. Throughout this report, when we discuss RVA, we are referring primarily to functionality within the plugin we created. When we discuss ParaView-RVA, we are referring to the combined functionality of ParaView with our RVA plugin.

Project Task Overview

The RVA project consisted of 15 tasks. The main emphasis of each task is summarized here.

Task 1 focused on the development of a Project Management Plan (PMP) overseen and approved by the project's DOE Project Officer.

Task 2 addressed the establishment of the Project Advisory Committee (PAC) and development of the first deliverable, the overall software design and major feature list. The content of the software design and major feature list were developed in consultation with the PAC.

Task 3 focused on development of code for importing and organizing data to be used within RVA. Software platforms and file formats identified for this task included UTChem, Isatis, ArcGIS shapefiles, Eclipse—ASCII and Z-Map.

Task 4 included development of code and support for visualization of volumetric data from geologic framework and reservoir simulation models.

Task 5 focused on the planning and development of the RVA User Manual.

Task 6 was the public release of the Beta 0.1 version of the RVA software via the RVA web site.

Task 7 included development of functionality for visualizing fluid flow data from reservoir simulations, the implementation of geo-location information, the visualization of faults and the visualization of wells.

Task 8 was the public release of the Beta 0.2 version of the RVA software via the RVA web site.

Task 9 focused on functionality for improving detailed visualization and application of spreadsheet visualization capabilities.

Task 10 was the distribution of the Beta 0.3 release of RVA via the RVA web site.

Task 11 addressed the development of code for data mining within the modeling results.

Task 12 was the distribution of release version 1.0 of the RVA plugin.

Task 13 included additional work on the data mining coding and functionality.

Task 14 was the distribution of release version 2.0 of the RVA plugin.

Task 15 focused on technology transfer, including attendance at professional meetings and presentation of 2 workshops promoting the use of the RVA software.

Primary Feature List

One of the first tasks within the project was, based on consultation with the PAC, the development of a detailed listing of high-priority features and functionality that we would address within the project. This effort resulted in the Prioritized Feature List for RVA Software (Table 1), that was provided to our DOE Project Manager in August 2011. This feature list presented key functionality according to 3 major themes, geologic framework model analysis, reservoir simulation analysis, and integrated environment controls.

Prioritized Feature List for RVA Software

Geologic framework model analysis

Import capability for key formats (3.1, 3.2) Isatis, UTCHEM, Petrel, Eclipse,
Custom tools and tutorials for visualization and analysis of multiple geologic properties using a range of visualization tools (4.1, 4.2)
Visualization of fault planes, including labeling (7.4)
Visualization of wells: both geophysical logs and geologists logs (7.5)

Reservoir simulation analysis

Import capability for key formats (3.1, 3.2)
Fluid property visualization, querying and analysis (e.g., fraction, temperature, pressure) (4.1, 4.2)
Flow line visualization (7.1)
Animation of flow lines (7.2)
Level of Detail for flow line visualization (9.1)

Integrated environment controls

Projection and coordinate system recognition and re-projection capability (7.3)
Visualization of spreadsheet interface for multiple, synchronized windows (9.2)
Visualization of x-y plots of at wells or at specific locations (4.1, 4.2)
Focus + Context interface for handling high-res data (9.3)
Capability to export data between spreadsheets and sub-windows (11.1)
Advanced feature detection capabilities (11.2)
Implementation of correlation mining capabilities (13.1)
Extension of mining capabilities across multiple time steps (13.2)

Table 1. Prioritized feature listing for RVA, by major theme, with relevant task numbers listed in parentheses.

Together with the PMP, the Prioritized Feature List was used to help guide and prioritize our efforts during the project.

Comparison of Accomplishments vs Goals

This analysis will be conducted using the categories and listing from Table 1 (Prioritized Feature List for RVA Software).

Geologic Framework Model Analysis

1. Import capability for key formats, including Isatis, UTCHEM, Petrel, Eclipse. We were able to successfully implement robust import capabilities for Isatis and UTChem. Isatis functionality was implemented easily because the vendor, Geovariance, offered free access to their software development kit. UTChem functionality was developed by RVA programmers because the output from UTChem was in open, documented ASCII files. However, because of the proprietary, fee-for-license nature of software development kits from Landmark and Schlumberger, we were unable to create import capabilities to read the native file formats from these large simulation platforms. Researchers working with PAC member, Dr. Robert Ritzi at Wright State University, were able to import a very large geologic framework model and fluid flow simulation from Eclipse, using the Eclipse ASCII exporting option. The images generated from these files, within ParaView-RVA, were used in an article published in GeoSphere (Gershenzon et al. 2015).
2. Custom tools and tutorials for visualization and analysis of multiple geologic properties using a range of visualization tools. This was completed as planned. The User Manual highlights many different ways to visualize and analyze reservoir simulation results using RVA. This report contains several images also highlighting the RVA capabilities.
3. Visualization of fault planes. We were able to successfully integrate fault planes from within Isatis project files. Reservoir simulation code, UTChem did not have fault planes as a recognized component of that software. Due to limitations in time, we were unable to develop code for a more robust and generic fault file format.
4. Visualization of wells, both geophysical logs and geologist's or driller's logs. We were able to implement visualization of geologist's and geophysical wireline logs from within Isatis. UTChem did not support well information describing either geologist's logs or geophysical wireline log data. UTChem well data was focused on well pumping history and associated fluid information. We were able to implement visualization and analysis of the UTChem well position and history files.

Reservoir simulation analysis

1. Import capability for key formats. See #1 above. We were able to develop import capabilities for UTChem 9.x simulation results. A colleague was able to import Eclipse simulation results after using the ASCII export functions within Eclipse.
2. Fluid property visualization, querying and analysis. This was implemented successfully. This is demonstrated in the User Manual, the web site gallery, and in Figure 4 of this report.
3. Flow line visualization. We were able to show how to take velocity information from UTChem, create flow lines, visualize these flow lines with a user-defined level of detail, and to animate these flow lines using standard ParaView filters and interface controls. This process is outlined it the User Manual.

Integrated environment controls

1. Projection and coordinate system recognition and re-projection capability. During the course of the project we discussed this feature with colleagues and PAC members. Through those

conversations, we determined that projection capabilities were not a highly valued function. Due to the difficulty in integrating this functionality seamlessly into ParaView-RVA, and due to the availability of re-projection capabilities in several other commonly used software packages, we dropped this from our prioritized feature list.

2. Visualization of spreadsheet interface for multiple, synchronized windows. This functionality is available within ParaView and additional functionality was added as this project went on. We demonstrate how to use the ParaView functionality within the User Manual. Additional instruction is available via ParaView documentation web pages.
3. Visualization of x-y plots at wells or specific locations. This was completed. We have demonstrated, within the User Manual, the inherent functionality within ParaView for creating x-y plots, using UTChem well history information.
4. Focus + Context interface for handling high-resolution data. A tool for evaluating the context of a given view was implemented within RVA. This tool creates a new 3-D window and using a wireframe of the original model, identifies the camera position for the original view.
5. Advanced feature detection capabilities. CloudRunner correlation analysis tool was developed to meet this feature requirement. CloudRunner is described below.
6. Implementation of correlation mining capabilities. This feature requirement was met within the CloudRunner correlation analysis tool.
7. Extension of mining capabilities across multiple time steps. The CloudRunner correlation analysis tool provides analysis of individual time steps, and can be applied sequentially to analyze multiple time steps in a fluid flow simulation.

Development of the RVA User Manual

One of the primary goals of this project was the creation of a User Manual that would provide significant assistance to users with a range of expertise in manipulation of software, particularly visualization software. The selection of ParaView as the base on which we developed RVA was due to the significant code base and functionality that ParaView provided and the design-goal of extendibility that was also a major element within ParaView. Based on discussions during the kick-off meeting at DOE –Houston headquarters, and on conversations with PAC members, we determined that the User Manual would have to provide some instruction in the use of ParaView, as well as guide users on the application of ParaView-RVA to manipulate their geologic data sets and reservoir model results.

The original proposal and budget called for hiring a technical editor to assist with the User Manual. However, we were unable to find a qualified editor to work with us on this project. To ensure the readability and overall utility of the User Manual, we consulted frequently with technical editors on staff at the ISGS. Guidance from the editors was used to develop a consistent style and strategy for the RVA User Manual. Because of the multi-window environment that is ParaView, and the widespread use of menus and buttons, we adopted a style within the User Manual that relied on both screen captures and a custom short-hand notation to instruct users on the specific panels, tabs, commands and parameters that needed to be used for a given effect. In addition, we assumed that the users of the manual were geoscientists and engineers with exposure to various aspects of the modeling and simulation workflow, but no presumed experience with ParaView or other visualization software. Comments from the PAC suggested these standards were effective in making a readable and user-friendly User Manual.

Creation and use of RVA-related web sites

To make the RVA plugin more discoverable, to explain the goals of the project, provide some example visualizations that are feasible with RVA, and to simplify the accessibility of the software, User Manual and demonstration data sets, we created a project web site (http://rva.cs.illinois.edu). This web site will remain operational for the long-term future.

RVA was designed and intended to be an open-source application, meaning that the original source code was available for reuse with proper citation, and that the main RVA code base itself was extendible by any interested party. To further simplify the accessibility of the code and to ensure that further development occurred within the standard open source paradigm for community code development, we also created a GitHub site dedicated to RVA (https://github.com/shaffer1/RVA). The GitHub site will also be maintained for the long-term future, and provides users access to executable versions of the code, source code, documentation of code development history, demonstration data sets and the current version of the User Manual.

Development of file import utilities

As outlined in the Prioritized Feature List (Table 1), RVA was originally designed to import a wide range of file formats, including those of the major geologic framework and reservoir simulation modeling environments from Halliburton and Schlumberger. To simplify integration of model results from the major proprietary software packages into RVA, it is necessary to access the Software Development Kits (SDKs) that the software vendors distribute to allow other applications to directly read their native file formats. Without access to the SDKs for each major package, users would have to export the complex modeling results into an ASCII file format. Because these software modeling packages are so complicated, exporting of results is confusing and can be difficult. Unfortunately, upon contact with representatives of Halliburton and Schlumberger software divisions, we found that these vendors charged significantly for access to their SDKs, making it impossible for us to use them in this project. We were able to find one vendor that did provide free access to their SDK. Geovariances, the maker of the industry-leading geostatistical simulation software, Isatis (http://www.geovariances.com), provides their SDK for Isatis to any developers via their web site. During our kick-off meeting, DOE staff expressed a strong interest in a chemical flooding simulation package developed out of the University of Texas-Austin, called UTChem. We contacted Dr. Mojdeh Delshad, then coordinator of UTChem development, who expressed an interest in our program and committed to provide support to our programmers to help us develop file readers for the various UTChem output files (Figure 1). In addition to Isatis and UTChem, we also found that the file formats for ArcGIS shapefiles and the Z-Map file formats were well documented and openly accessibly, allowing us to develop input tools for these formats. Following discussions with our project team and PAC members, we determined that these formats were sufficient to help us develop the appropriate tools for RVA and would allow us to demonstrate the functionality that ParaView-RVA could provide. If additional file formats were desired in the future, we could explore ways to access those and develop the necessary reader applications. The User Manual describes how to read in all of these file formats.

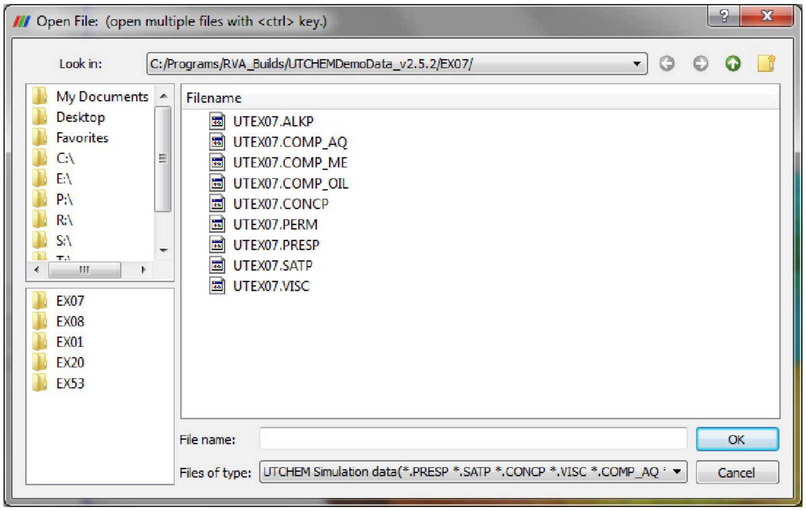

Figure 1. Open File window highlighting built-in recognition for UTChem simulation data.

Implementation of well visualization

Wells and the data that come from them are key components of oil fields and as such provide key information to support both geologic framework and reservoir simulation modeling. Integration of well data into ParaView-RVA was seen as critical for its success. Within both Isatis and UTChem, wells are formal data types. As expected, however, well data are recorded and managed differently in each software environment. The Isatis SDK, GTXserver, provides access to native Isatis project files and allows ParaView-RVA to read in well data as native data types (Figure 2). UTChem data also manages well data in separate files, and a special well-reader code was needed to support these (Figure 3).

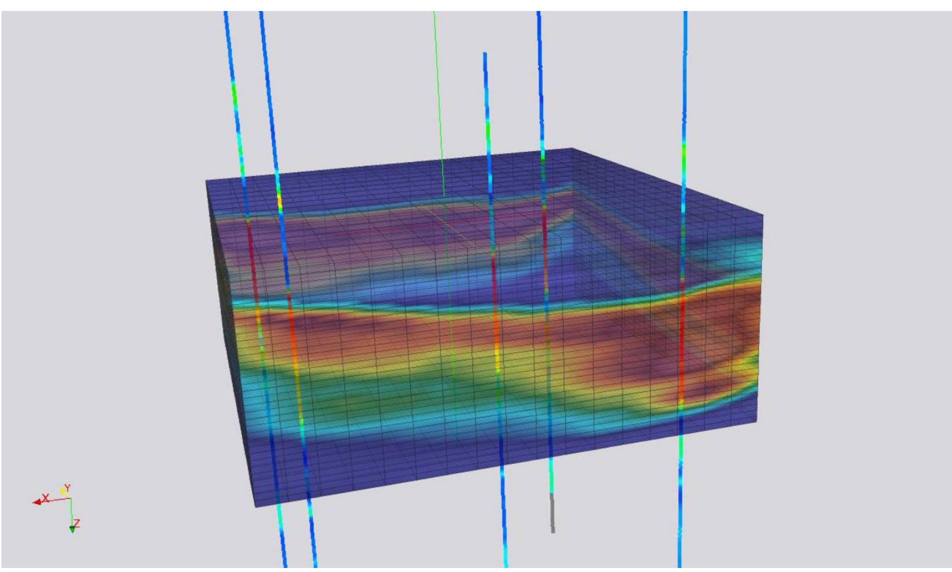

Figure 2. Isatis demonstration model of permeability field with wells shaded on normalized SP values.

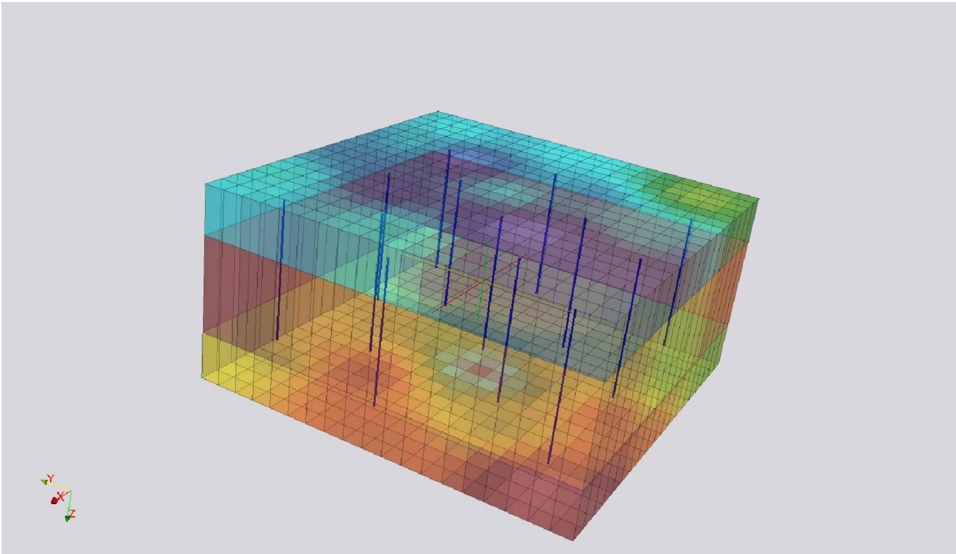

Figure 3. UTChem simulation of permeability field with wells visible.

Implementation of connectivity functionality

Connectivity or partitioning of reservoir properties, both structural and fluid, can be important to understanding the effectiveness of a given pumping or EOR strategy. The Connected Threshold and Connected Threshold with Custom Source tools were developed to assist in understanding reservoir connectivity within RVA. The Connected Threshold tool identifies sets of contiguous cells within a reservoir model that satisfy a user-specified condition (Figure 4). The Connected Threshold with Custom Source tool identifies sets of contiguous cells that also connect with a specified well or set of wells. Both tools allow for complex conditionals to be framed through the use of two variables and a range of Boolean operators. These tools allow for a sophisticated analysis of reservoir framework or fluid simulation model properties, and, as appropriate, provide a visualization of partitioning over time from transient flow model results.

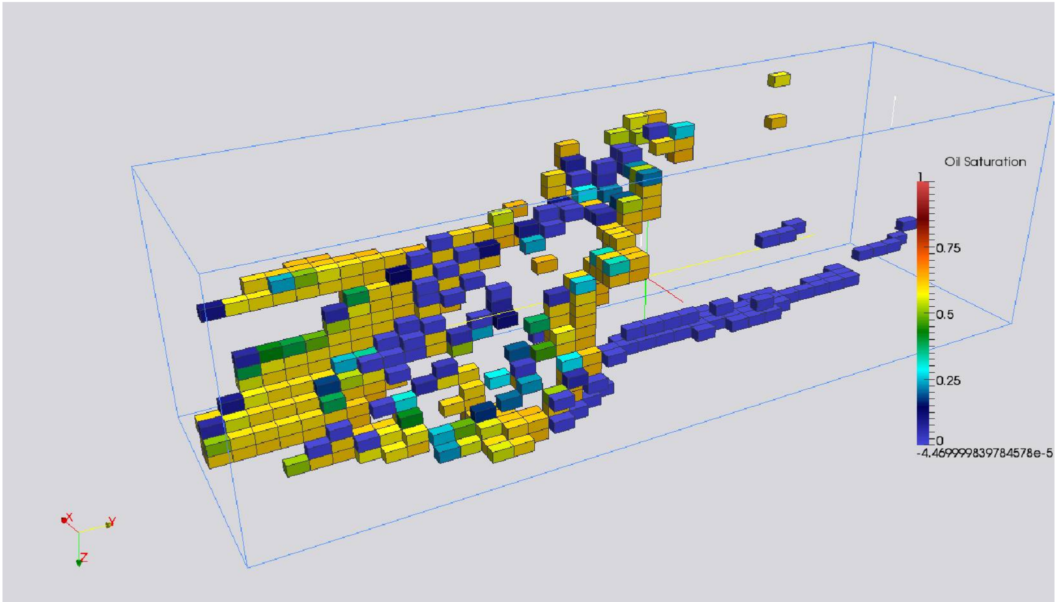

Figure 4. UTChem simulation Ex20 with reservoir connectivity evaluated based on oil saturation between 0.001 and 0.64.

Implementation of fault visualization

Geologic reservoirs are often faulted, and the geometry of these faults can play a major role in controlling the fluid flow within the reservoir. While fluid flow simulators do not always directly integrate information on the distribution of faults, visualization of faults can be significant in helping understand the results of reservoir simulations and better understand management options for the reservoir. Isatis allows for the specification of faults within their projects. To demonstrate the value of fault visualization, we developed code to ingest faults from Isatis projects into ParaView-RVA.

Implementation of geo-location functionality

Geologic framework and reservoir simulation models are increasingly built with hundreds of thousands to millions of cells. To assist in the analysis of these modeling results, it can be helpful to locate specific cells or regions of the models. The geo-location tool in RVA was designed to allow a user to center the view to a specific cell or location within a model. Locations can be specified either as index values, x, y, z, or fractional positions (e.g., 0.3, 0.42, 0.1), or with real world coordinates (Figure 5). Once entered, the view is immediately centered to that coordinate location, providing the user a bit more control over the perspective they have on the model results.

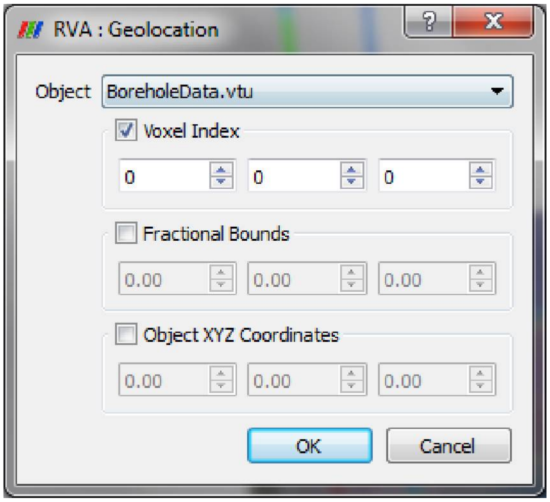

Figure 5. Geolocation window.

Implementation of data mining capabilities

Data mining capabilities were developed for this project through the development of a new application, CloudRunner. CloudRunner was developed in consultation with PAC members, particularly Dr. Mojdeh Delshad at UT-Austin. The large output file size of EOR simulations can be very slow and computationally expensive to analyze. CloudRunner is designed to shorten computation times and make the analyses more tractable, by analyzing multiple simulation realizations, and applying data mining algorithms using common MPI-based parallel or distributed computer environments (i.e., multi-core workstations, clusters, commercial cloud environments). CloudRunner calculates the mean, variance, standard deviation and Pearson's correlation coefficient for specific attributes within a single simulation or across multiple simulations. CloudRunner also computes reservoir connectivity, based on user-specified thresholds, for any simulated reservoir property. CloudRunner has been tested on UTCHEM simulation results. Based on recommendations from Dr. Delshad, CloudRunner has been implemented as a stand-alone application, making it easier to use and more flexible in how it can be deployed. The full source code for the CloudRunner data analysis software is available on the web at https://github.com/dmcwhe2/CloudRunner.

Technology Transfer

We addressed the technology transfer obligations listed within our PMP through a suite of efforts. In the Spring of 2012, we gave a presentation at the Spring meeting of the Illinois Geological Association, which is a local professional organization made up of geologists and upstream petroleum companies that operate within the Illinois Basin. Because this presentation was made early in our development cycle, at the meeting we presented the design goals of the project but did not have an operational product to showcase. We talked to attendees about their interest in the software and interest in participating as we developed the RVA plugin. While there was some interest in the software, most interest was in the development of simple reservoir volumetrics calculations. There was little interest in the visualization of geologic framework or reservoir simulation models, as most of the small companies and representatives at this meeting did not create geologic framework models or run reservoir simulations. Another PMP commitment was to attend the National AAPG conference. We attended the 2015 AAPG Conference in Denver, Colorado and met with a range of participants. We targeted both small- and mid-sized companies and Universities. We received the best support for our software by University faculty and students, and reception from small- or mid-sized E&P company representatives was poor. Most company representatives had established workflows and were not very interested in new software. It seems likely that a focused marketing or promotional effort might get some exploratory downloads of the software, and this is something to consider for future efforts. Future efforts might try an exhibitor booth at a national conference with multiple computers for demonstration purposes. Given the bug-fixes we were working with at the time, we couldn't commit to an exhibitor's booth for this conference. Finally, we conducted 2 workshops at the end of the project to demonstrate the software. The first workshop was on the University of Texas –Austin campus, where we presented to campus and local industry representatives. We worked with faculty in the Department of Petroleum and Geosystems Engineering to advertise and host this workshop. This workshop was attended by approximately 20 people and the reception was very good. There was interest expressed in using the RVA plugin and in possibly developing it further to work with other reservoir simulators that they work with. The second workshop was a 2-part webinar via WebEx. To advertise and promote this, we contacted over 100 individuals via direct email. Most of these individuals were exhibitors at the 2015 AAPG Conference in Denver, and included Universities and upstream support companies. Unfortunately, these webinars were sparsely attended, with approximately 10 different people participating over the 2 parts.

Publications

Collaborations with Dr. Robert Ritzi and Dr. Naum Gershenzon, at Wright State University, led to the analysis of a large EOR simulation, within a large, high-resolution sedimentary framework model that was run in Eclipse. The collaboration from this effort led to the following publication:

Gershenzon, ,N.I., M.R. Soltanian, R.W. Ritzi, D.F. Dominic, D. Keefer, E. Shaffer, B. Storsved, 2015. How does the connectivity of open-framework conglomerates within multi-scale hierarchical fluvial architecture affect oil-sweep efficiency in waterflooding, Geosphere, v. 11, no. 6, 18, pp: 1-18, doi:10.1130/GES01115.1.

Networks and collaborations fostered

Through this project, we have established connections with Drs. Mojdeh Delshad and Kamy Sepehrnoori, from the Petroleum and Geosystems Engineering Department at the University of Texas – Austin, related to our support for the UTChem simulator and the workshop we held at UT-Austin for this project.

We have increased collaborations with Dr. Robert Ritzi, Wright State University who served on our PAC and has since increased his collaboration with the ISGS through his participation in a US DOE-funded Energy Frontiers Research Center project, led by the ISGS at the University of Illinois.

Conclusions and Recommendations

ParaView-RVA has proven to be a success, with new tools and custom instruction via a User Manual that provide new and significant assistance in improving the capabilities for visualizing and analyzing complex geologic modeling and reservoir simulation modeling results, including multi-fluid, multi-property interrogation and visualization. The built-in functionality of ParaView provides the capacity to process and visualize data sets ranging from small models on local desktop systems to extremely large models created and stored on remote supercomputers. The RVA plugin that we developed and the associated User Manual provide improved functionality through new software tools, and instruction in the use of ParaView-RVA, targeted to petroleum engineers and geologists in industry and research. The RVA web site (http://rva.cs.illinois.edu) provides an overview of functions, and the development web site (https://github.com/shaffer1/RVA) provides ready access to the source code, compiled binaries, user manual, and a suite of demonstration data sets. This simplifies user-community support of RVA through an industry standard community code development platform. The demonstration data sets also allow users the opportunity to explore the functionality of ParaView-RVA without having their own data sets in hand.

Key functionality has been included to support a range of reservoirs visualization and analysis needs, including: sophisticated connectivity analysis, cross sections through simulation results between selected wells, simplified volumetric calculations, global vertical exaggeration adjustments, ingestion of UTChem simulation results, ingestion of Isatis geostatistical framework models, interrogation of joint geologic and reservoir modeling results, joint visualization and analysis of well history files, location-targeted visualization, advanced correlation analysis, visualization of flow paths, and creation of static images and animations highlighting targeted reservoir features.

Further extension of ParaView-RVA could be designed to provide even more support to energy-sector geoscience and engineering users. Additional functionality could include development of file readers for other popular geologic framework and reservoir simulation packages, development of new open file formats to better accommodate volumetric, transient simulations, to better interact with a range of GIS file types, and to better integrate functionality around wells and well data. Additionally, improved instruction, via the User Manual, on the use of the built-in ParaView Python calculator, with demonstrated examples of common problems, could provide important advice to improve the ease of use for new users.

It is important to note that through our efforts on this project, we found that using ParaView for sophisticated analyses requires a bit more effort and expertise than is suggested within the ParaView web sites and documentation. Programming experience in Python or C++ also provides an advantage in simplifying complex analysis within ParaView. We found that the code base of ParaView was sufficiently complex that extending ParaView with new code, as was done for this project, is much more effective if the programmers have prior experience programming within ParaView. We were able to demonstrate that, with programmers who have sufficient experience, ParaView can be a productive and beneficial software package to work with, and development of custom tools can provide significant assistance to targeted user groups.

In discussions with geologists and petroleum engineers in small- to mid-sized upstream petroleum companies, we found a reluctance to explore ParaView-RVA. These responses could be due to several factors. The development schedule established in our PMP, resulted in a product with limited functionality until late in the project life. This limited the utility of demonstrations or presentations at professional conferences or other venues. Future software development projects of this nature should consider setting up work plans so demonstrations and roll-out events are scheduled for after significant features have been created and tested. In addition to participating at national conferences, time and resources should be placed on creating brochures and demonstration videos that highlight functionality of interest to target groups; these can be made available both in-person and via project web sites. Consideration should also be given to obtaining exhibitor space at national conferences to provide hands-on demonstration opportunities with target user groups.

In contrast to relatively cool reception from petroleum company representatives, research groups in Universities and State Geological Surveys were generally very receptive and interested in exploring ParaView-RVA. These groups often are much more focused on expanding research capabilities and were receptive to learning new software if it provided needed functionality. They are also routinely involved with graduate students who often have time available for exploring new software and programming skills that can help adjust software functionality and data structure fit. Future software development projects could take advantage of this more directly by enlisting Universities into feature testing and advisory committee responsibilities.

Finally, and in retrospect, it is worth noting that the timing and number of the Go-No Go Decision Points in this project created logistical problems for efficient completion of the project goals. In development of our PMP and with the direction of our original project manager, Virginia Weyland, we scheduled 5 Go/No Go Decision Points to be in sync with incremental releases of the RVA software. While this seemed appropriate at the time, it turned out to be very problematic. In retrospect, with our ambitious set of objectives, the final schedule did not allow sufficient time for feature testing and robust bug identification prior to each version release and Go/No Go decision point review. While we were able to meet our deadlines within this schedule, it resulted in a number of bugs that were not identified until after version releases, and that needed to be corrected during the no-cost extension periods. In addition to these logistical problems, our prior inexperience with Go/No Go decision points resulted in a misunderstanding of the project workflow during the decision-point review period. As a non-profit agency, the University does not have surplus revenue that could be charged if an unfavorable review led to a project discontinuation. Accordingly, University officers instructed us to discontinue working on the project during each Go/No Go review period. This resulted in 5 work stoppages and necessitated movement of contract-funded staff to other projects while the decision point was reviewed. Future software development projects would be better structured by reducing the number of Go/No Go decision points, and setting them significantly after software releases, thereby allowing sufficient time for feature testing and bug identification, and creating fewer disruptions of workflow and progress on objectives.

List of Acronyms and Abbreviations

CS – Department of Computer Science at the University of Illinois at Urbana-Champaign

ISGS – Illinois State Geological Survey, a division of the Prairie Research Institute at the University of Illinois at Urbana-Champaign.

NCSA – National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign

RVA – Reservoir Visualization and Analysis plugin for ParaView 3.14. (http://www.rva.cs.illinois.edu)

PAC – Project Advisory Committee

PMP – Project Management Plan

SDK – Software Development Kit, a suite of software development tools that allows the creation of applications to work with a specific software package (source: Wikipedia).

UIUC – University of Illinois at Urbana-Champaign

VTK – Visualization Toolkit, an open source library for management, visualization and analysis of 3-D objects (http://www.vtk.org)