Environmental Modelling & Software 85 (2016) 11–26

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

A new open source platform for lowering the barrier for environmental web app development

Nathan R. Swain [a, *], Scott D. Christensen [b], Alan D. Snow [c], Herman Dolder [a], Gonzalo Espinoza-Dávalos [d], Erfan Goharian [e], Norman L. Jones [f], E. James Nelson [f], Daniel P. Ames [f], Steven J. Burian [g]

[a] Aquaveo, LLC, Provo, UT, USA
[b] Information and Technology Laboratory, United States Army Engineer Research and Development Center, Vicksburg, MS, USA
[c] Coastal and Hydraulics Laboratory, United States Army Engineer Research and Development Center, Vicksburg, MS, USA
[d] Department of Integrated Water Systems and Governance, UNESCO-IHE, Delft, The Netherlands
[e] Department of Land, Air and Water Resource, University of California, Davis, CA, USA
[f] Civil and Environmental Engineering, Brigham Young University, Provo, UT, USA
[g] Civil and Environmental Engineering, University of Utah, Salt Lake City, UT, USA

ARTICLE INFO

Article history:
Received 12 February 2016
Received in revised form
2 August 2016
Accepted 8 August 2016

Keywords:
Tethys platform
Web app development
Environmental
Decision support
Hydrologic modeling

ABSTRACT

The interactive nature of web applications or "web apps" makes them a well-suited medium for conveying complex scientific concepts to lay audiences and creating decision support tools that harness cutting edge modeling techniques and promote the work of environmental scientists and engineers. Despite this potential, the technical expertise required to develop web apps represents a formidable barrier—even for scientists and engineers who are skilled programmers. This paper describes four hurdles that contribute to this barrier and introduces an approach to overcoming these hurdles. We present an open source implementation of this approach, a development and hosting environment for environmental web apps called Tethys Platform. Several case studies are provided that demonstrates how the approach, as implemented within Tethys Platform, successfully lowers the barrier to web app development in the environmental domain.

1. Introduction

Hydrologic and other types of environmental model simulations are often used in decision making to estimate and analyze watershed responses to specific scenarios (Bhuyan et al., 2003; Goodrich et al., 2008; Lam et al., 2004; Miller et al., 2007; Santhi et al., 2006). However, typical stakeholders and decision makers do not have the technical expertise required to properly configure a simulation for a particular scenario. The process becomes even more daunting for physics-based hydrologic models, because of the challenges of data collection and management of large spatial and temporal datasets.

Environmental web applications or "web apps" can overcome many of the challenges of using hydrologic simulations in decision-making (e.g.: Demir and Krajewski, 2013; Goodrich et al., 2008; Kulkarni et al., 2014; Sun, 2013). In the context of this work we define an environmental web app as a narrowly-focused, web-accessed application for performing common tasks related to environmental modeling. In a web environment, environmental web apps can be hosted on a remote server that can be accessed simultaneously by multiple users via a web interface. This eliminates the need for the end user to procure and maintain the high performance hardware required by the models, deal with issues related to software installation and operating system incompatibilities, or monitor and install software updates. All that is needed to use the web app is an internet connection and a web browser.

Despite the potential of web apps for promoting the work of environmental scientists and engineers, the technical expertise required to develop them represents a significant barrier for would-be developers whose primary background is environmental modeling. The barrier can be characterized by several hurdles a novice developer would need to overcome to successfully develop an environmental web app.

* Corresponding author.
E-mail address: nswain@aquaveo.com (N.R. Swain).

http://dx.doi.org/10.1016/j.envsoft.2016.08.003
1364-8152/

One hurdle is the need to discover and select software packages that address the spatial needs of environmental web apps. These spatial needs can be addressed using existing free and open source software (FOSS) for geographic information systems (FOSS4G), but the abundance of FOSS4G that are available can be overwhelming to new developers. For example, the Open Geospatial Consortium (OGC), an organization that creates standards for GIS software, publishes a database of over 800 registered products (Open Geospatial Consortium, 2015). Moreover, of the 34 geospatial software packages promoted by the Open Source Geospatial Foundation (OSGeo) 14 are for web mapping applications (includes fully adopted projects and those in "incubation" mode). Swain et al. (2015) addressed this challenge of navigating the daunting open source technology space by performing a review of the state-of-the-art FOSS4G and FOSS for web software packages that have been used in earth science web apps in the literature.

There are many benefits of using FOSS4G packages to address the needs of environmental web apps, but using FOSS4G packages often requires orchestrating the use of more than one package to achieve the desired functionality. FOSS4G packages tend to be more narrowly focused in terms of functionality than their proprietary counterparts. Whereas proprietary software vendors typically offer a wide variety of GIS functionality (e.g. web mapping services, geoprocessing, and spatial storage) in a single software package, FOSS4G packages tend to focus on a single category of functionality (Steiniger and Weibel, 2010). Consequently, creating an environmental web app using FOSS4G usually requires the developer to synthesize several packages and orchestrate their use via code.

Another hurdle that developers encounter is caused by the multi-lingual nature of web app development. Developing a dynamic, interactive web app requires the use of HTML, CSS, and JavaScript for creating the user interface; a scripting language such as PHP, Python, or Ruby for handling logic on the server; and structured query language (SQL) for interacting with a database. Successful web app development also requires the use of a software architectural pattern such as model-view-controller (MVC) or some variant to prevent the source code from becoming unmanageable (Buschmann et al., 1996; Feng et al., 2011; Gamma et al., 1995; Jansson and Moon, 2001; Mason et al., 2014; Walker and Chapra, 2014). We classify the above-mentioned challenges into four major hurdles: (1) the software hurdle, (2) the orchestration hurdle, (3) the web development hurdle, and (4) the deployment hurdle (summarized in Table 1).

Our primary objective in this work was to demonstrate an approach for lowering the barrier for environmental web app development so as to make it a more viable medium for environmental scientists and engineers who have some scientific programming experience. Our approach lowers the barrier to water resource web development by addressing each of the four hurdles by providing: (1) open source software that meets the spatial and computational capabilities commonly required for environmental modeling; (2) a programmatic means to use each of the recommended software tools in a single programming language; (3) a reduction of the web development skills required to develop web apps; and (4) a web-safe mechanism for deploying the finished web apps that is flexible enough to work on the most common hardware (i.e. university cloud, commercial cloud, private data centers). As a means of illustrating this approach, we present an implementation of this approach, a development and hosting environment for environmental web apps called Tethys Platform.

The remainder of this paper is organized as follows. A description of the three primary components of Tethys Platform and their design is presented in Section 2. The capabilities of Tethys Platform are demonstrated in Section 3 with descriptions of several web apps that were developed. A detailed discussion on how the barrier to web app development has been successfully lowered by Tethys Platform is presented in Section 4.

2. Software description

Tethys Platform is a development and hosting environment for environmental web apps. Although it aims to lower the barrier to water resource web app development, web apps developed using Tethys Platform are created programmatically—not using a graphical drag-and-drop type editor. It is targeted at motivated scientists and engineers who have some scientific programming experience, but not necessarily web development experience. We assume that users of Tethys Platform recognize the value of disseminating their work through web app medium, but are either daunted by the prospect of learning web development and/or have little patience for the sometimes tedious task of developing visually appealing web user interfaces. We selected Python, an all-purpose scripting language (Python Software Foundation, 2016) as the programming language of Tethys Platform, because it is relatively easy to learn and it has gained popularity among scientists and engineers in recent years (Millman and Aivazis, 2011; Oliphant, 2007).

Tethys Platform consists of three major components: Tethys Software Suite, Tethys Software Development Kit (SDK), and Tethys Portal. Each component was designed to address one or more of the four hurdles and effectively lower the barrier for development. Tethys Software Suite overcomes the software hurdle by providing suite of 3rd party FOSS and FOSS4G software tools to address many of the common needs encountered in environmental web app development. Tethys SDK addresses both the web development hurdle and the orchestration hurdle by providing a Python MVC framework for streamlined development of the web apps and Python APIs that allow programmatic control over each of the software suite component. Tethys Portal overcomes the deployment hurdle, by providing the primary runtime environment for Tethys Platform web apps. Fig. 1 summarizes the major components of Tethys Platform, which are discussed in more detail in this section.

2.1. Tethys Software Suite

Tethys Software Suite is the component of Tethys Platform that provides access to resources and functionality that are commonly required to develop environmental web apps. The primary motivation of creating the Tethys Software Suite was to address the software hurdle discussed previously. Some of the more specialized needs environmental app must provide arise from the spatial data components of the models that are used in the apps. Distributed

Table 1
Summary of the four challenges identified.

Challenges	Description
Software hurdle	Selecting from many available FOSS and FOSS4G to provide the spatial and computing capabilities required by environmental web apps.
Orchestration hurdle	Synthesize multiple FOSS and FOSS4G to provide a broad range of capabilities.
Web development hurdle	Learn multiple languages such as HTML, CSS, JavaScript, Python, PHP, Ruby and code management approach such as model-view-controller.
Deployment hurdle	Deploy the completed web apps in a safe and secure way on varying hardware and data center environments.