Re-using Engineering Tools: Engineering SaaS web application framework

R. Klinc[1], I. Peruš[2] and M. Dolenc[3]

The Institute of Structural Engineering, Earthquake Engineering and Construction IT (IKPIR), University of Ljubljana, Faculty of civil and geodetic engineering, Jamova 2, 1000 Ljubljana, Slovenia
[1] rklinc@itc.fgg.uni-lj.si 2 iperus@ikpir.fgg.uni-lj.si, [3] mdolenc@itc.fgg.uni-lj.si

ABSTRACT

While early days of structural analysis (and other engineering) software have been dominated by self-developed and self-maintained programs and applications, most of these tools have lately been abandoned in favor of out-of-the-box commercial software packages, usually due to the inability to cope with the advancements in the computer industry from both software and hardware perspective. Even though the computational engines of these software tools and applications are still sound, the pre- and post-processors are outdated and not suitable for the modern engineering use. On the other hand, the last decade offered simple, intuitive, effective and widely accessible services of the modern web resulting in the fusion of services blurring the boundaries between desktop computers, laptops and other consumer devices. Consequently, more and more applications are developed as web services, moving from desktop environment to the browser. This paper presents the architecture, development and deployment of the web application framework that can transform console standalone tools into fully functional web applications. The framework was developed following the principles of the service-oriented architecture using cloud computing guidelines and the software-as-a-service model of deployment. The use of modern web programming principles and techniques provides the ability to use the framework on a wide range of consumer devices while the underlying high performance/throughput computing system ensures scalability, flexibility and performance. Last but not least, two case applications utilizing the proposed framework are presented.

INTRODUCTION

Since the introduction of computers into the construction industry in the

1960s, the architecture, engineering and construction (AEC) industry went through several phases of technology adoption. Turk and Cerovšek (2001) noted that computers were first used to power special purpose programs (such as numerical analysis and similar) that allowed engineers to solve larger models with higher precision. The second phase was characterized by computers replacing drawing boards; however, the real change began with the third phase in the mid-1980s when computers became affordable, accessible and ubiquitous in the engineering profession. That was the tipping point for the use of computers in AEC – the complexity of mathematical models representing real buildings was not limited by the size and computer processing power anymore. Instead, the only limitations were the available time, available financing and last but not least, engineers' imagination.

Among the outcomes of this early engineering-oriented computing period were the first widely available user-oriented production programs for solving engineering problems. Many of them were under development and constant improvement for decades, maintaining core of the system and developing pre- and postprocessors. Over the years, most of this in-house self-developed programming solutions lost battle against newly emerged commercial software. While most of the outdated engineering software solutions had strong and sound theoretical foundation, the problem was with the compatibility of the developed solutions with underlying software and hardware. In addition, fourth (introduction of the internet) as well as fifth (mobile consumer devices and the consumerization of information technology) phase of the adoption of computers in AEC industry brought higher expectations regarding user interfaces and user interactions with the software. This is where all-in-one developers of the special purpose software solutions that were not dealing with programming professionally could not cope with the advances in the computer and information technologies. Consequently, once common endeavor of writing in-house domain specific software programs has (almost) vanished.

In view of the facts that construction industry is conservative, that knowledge is built over time and that years and decades of knowledge, validations and experiences were built into software solutions that were abandoned due to reasons beyond the precision, accuracy, speed or price, this paper proposes a web-based application framework that enables the transformation of the console stand-alone tools into fully functional web applications. Furthermore, the application of the program for elastic dynamic analysis of multistory structures (EAVEK) as well as the application of NANDRAD, a solver kernel for multi-zone building energy performance simulation onto the developed framework, is presented.

BACKGROUND

To provide a better outlook on the research regarding the proposed framework, a brief review of the concepts, tools, and infrastructure used in developing the framework is presented in this section.

One of the concepts used in the development of many (if not all) of the modern web services is **cloud computing**. Cloud computing is a general concept that incorporates software as a service (SaaS), Web 2.0 and other technology trends, in which the common theme is reliance on the Internet for satisfying the computing needs of the users. It is often defined as "*a paradigm in which information is permanently stored in servers on the Internet and cached temporarily on clients that include desktops, entertainment centres, tablet computers, notebooks, wall computers, handhelds, sensors, monitors, etc*" (Hewitt 2008). The cloud is a metaphor for the Internet, based on how it is depicted in computer network diagrams, and is an abstraction for the complex infrastructure it conceals. It is a style of computing in which IT-related capabilities are provided "as a service", allowing users to access technology-enabled services from the Internet without the knowledge of, expertise with, or control over the technology infrastructure that supports them.

Another phenomenon that attracted a lot of attention in the last decade and had impact on our framework is **Web 2.0**. It is considered as a next step and a major evolution of the traditional web from both the technological and social perspective. The term has numerous definitions and more or less all of the authors agree that it is a trend, a perception of the direction the Web is heading, and not an object that can be created (Klinc et al. 2009). Web 2.0 cloud computing model together with other related technologies and business practices can provide the needed information and communication infrastructure that will enable AEC SMEs to compete with larger engineering companies (Klinc et al. 2010).

PROPOSED WEB BASED APPLICATION FRAMEWORK

Since most of the traditional early-age engineering applications follow the basic code of action (see Figure 1), the architecture of the proposed framework follows classic client-server architecture, separated to presentation and business logic layer, although the architecture is flexible.

The role of the presentation layer in the proposed framework is to interact with users. Its main function is to translate tasks and results to something which users can understand. In the proposed framework the user interface has to be able to receive input as well as provide the results of the framework application used. In order for the framework to be flexible, scalable and user friendly, to ensure cross-platform compatibility and to hide the complexity from the end-user, the framework presentation layer is disseminated via an internet browser, which handles web pages, encoded in (X) HTML language and generated by the web server on the layer of business logic. Calls between the user interface and the web server are both synchronous and asynchronous. For asynchronous calls, the Ajax web programming approach was applied.

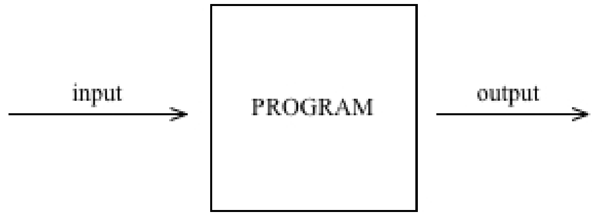

Figure 1: Code of action for traditional engineering applications.

The business logic layer is based on an Apache web server, running on a Linux platform. Scripts written using PHP programming language handle requests, process the input, interact with underlying applications, process output and prepare the resulting (X)HTML pages. Although the architecture of the framework is seemingly simple it can be further developed if necessary. In addition to expansion to third layer (data access), the non-trivial extensions are possible, especially regarding the scalability of the proposed network (connections with high-performance and high-throughput computing environments etc.).

CASE APPLICATION 1 – EAVEK

EAVEK is a program that was designed for the elastic static, dynamic and stability analysis of multistory structures, utilizing the theories of the 1^{st} and the 2^{nd} order and finite element method. The first version of the EAVEK was made in 1972, although the roots date back to 1971. Since then the program had been developed for decades. It was extensively used for design, research and education, especially in the region of former Yugoslavia and in the Peoples Republic of China (Fajfar, 1987; Kilar and Fajfar, 1997).

The EAVEK program was written in FORTRAN 77 (ANSI, 1978), general purpose programming language, mainly intended for mathematical computations in engineering and was initially developed for IBM PC compatible computers. While it was the most appropriate solution at the time, it is one of the main disadvantages of the program from today's point of view. Since the start of the EAVEK development, FORTRAN has gone through several iterations (FORTRAN 90, 95 2003 2008). Unfortunately, the problems with portability of the code as well as with incompatible extensions have not been sufficiently addressed. As a result, the development of the program stopped in early 1990s in spite of the wide and strong professional community. To re-facilitate the application in the construction industry, EAVEK has been applied to the proposed framework in three major steps:

1. The environment for running the program on a modern operating system had to be prepared. For this reason, Linux platform was chosen for its flexibility and variety of Fortran compilers.
2. Due to the computational complexity of the mathematical operations built in the program EAVEK had to be recompiled and extensively tested in order to verify the results.
3. The web environment had to be prepared as well as PHP pre- and post-processors had to be developed (Figure 2 shows the activity diagram and the data flow among the major components of the framework application).

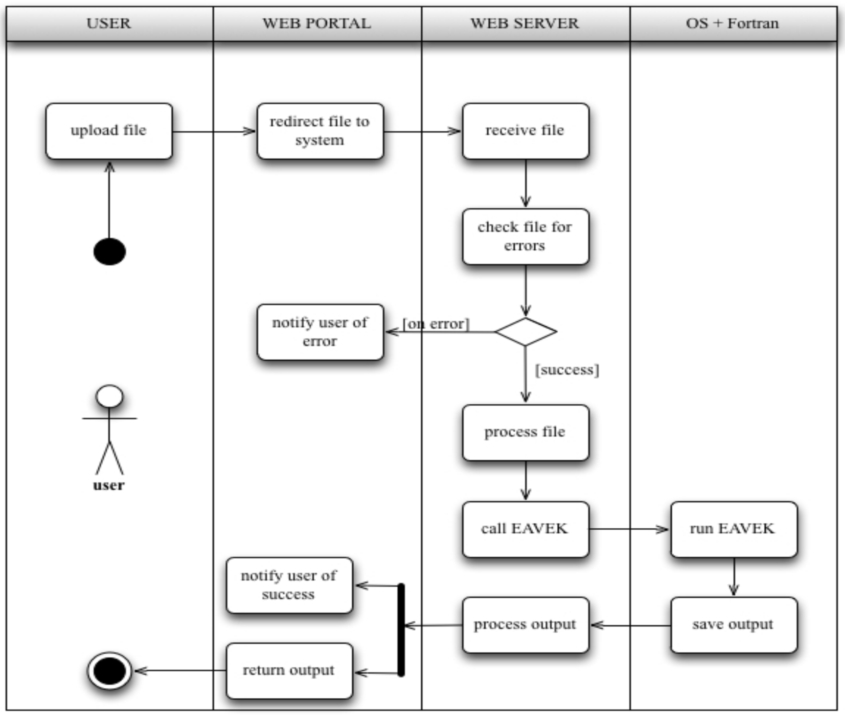

Figure 2: Activity diagram for EAVEK case application.

The application runs on the framework and is accessible through any standard web browser. With a view to ensure the accuracy and precision of the results, the application is currently under the extensive testing which is performed by the students in the educational process. Once the verification process is finished, the pre- and post-processors for the preparation and processing of the data will be enhanced. That is when the application will be presented to the general public.

CASE APPLICATION 2 – NANDRAD

In general, parametric studies are of high importance in science and in engineering. They usually consist of a number of repetitive, independent calculations that are therefore easily parallelized. Scientists and engineers involved in this type of work need a computing environment that delivers large amounts of computational power over a long period of time. Such an environment is called a high-throughput computing (HTC) environment (Dolenc and Dolšek 2008).

To address issues regarding high-volume parametric studies in engineering and to demonstrate the adaptability and scalability of the proposed framework it has been coupled with HTCondor, a batch queuing system for managing compute-intensive jobs. HTCondor provides a HTC environment that efficiently harnesses and makes the best use of all available resources while providing high throughput for jobs. In a typical usage scenario, a user submits a job to HTCondor, which queues and monitors the job and then presents the results when the job completes. HTCondor system also provides an application programming interface compatible with Distributed Resource Management Application (DRMAA). This enables developers to build end-to-end vertical solutions for specific end-user scenarios.

For this scenario, NANDRAD solver kernel for multi-zone building energy performance simulation has been applied to the enhanced framework (the activity diagram of the enhanced framework application is shown in Figure 3).

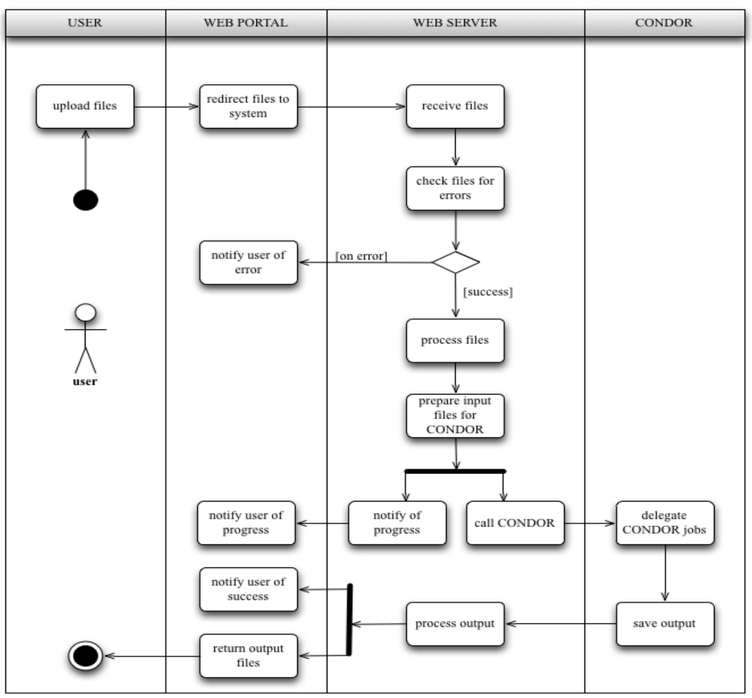

Figure 3: Activity diagram for NANDRAD case application.

NANDRAD is designed to perform transient solutions to energy balances in thermal zones and discretized construction elements. It provides also thermal comfort evaluation (e.g. indoor air temperature, operative room temperature, temperature of the inner surface of envelope elements). NANDRAD is a console application using files based inputs/outputs and has been applied to the proposed framework similarly to EAVEK with the addition of the HTCondor system utilization. The pre- and post-processors as well as the HTCondor plugin have all been developed using PHP. In addition to PHP and (X)HTML, AJAX programming techniques have been used for the user notification system.

As NANDRAD is available for Windows environment only, the application to the platform solved the problem of cross-platform compatibility. What is more, the proposed solution is scalable and significantly reduces the time for the execution of the performance simulations.

DISCUSSION

In this paper, an engineering web-based application framework that enables the transformation of the console stand-alone tools into fully functional web applications has been presented. Even though many researchers have worked on the web based frameworks, very few, if any, have reported results regarding the application of the framework for outdated and abandoned (for several reasons, none of them related to the overall performance or the professional value), domain specific software.

The objective of the proposed framework was twofold: first, to re-enable, re-use and re-introduce engineering applications that have been abandoned despite years of experiences and validations due to reasons associated with compatibility and user interfaces; second, to provide and enhance the existing console applications with a web based environment that is platform independent and can scale according to user and application needs. The main contribution of this paper is the description of the approach that enables re-use of the decades worth of experience and knowledge encapsulated in the abandoned and forgotten domain specific software.

CONCLUSIONS AND FUTURE WORK

Performances of the presented framework and its applications have indicated significant potential for usage in everyday engineering production environment. Among the advantages of the proposed solution are the adaptability, hidden complexity of the underlying system, cross-platform independence, advanced scalability, wide accessibility, single-point security and simplified maintenance. However, it should be emphasized that the applicability of the framework in the proposed form is limited to the console applications with text-based input/output.

As it is of most importance to provide correct and accurate output results, both framework applications presented have been extensively tested and validated.

In the next step, efforts will be oriented towards the evaluation of the proposed architecture and the behavior of the applied programs in real construction projects. To achieve this, additional work on the pre- and post-processors will have to be done in order to improve user interfaces and to offer applications that are rich, affordable and easy to use, and that fit well into the workflow of the AEC professional.

ACKNOWLEDGMENT

The work presented in this paper has been partially supported by the project ISES (Intelligent Services For Energy-Efficient Design and Life Cycle Simulation, http://ises.eu-project.info/) through EC's 7th Framework Programme, grant agreement (n° 288819).

REFERENCES

ANSI (1978). “Ansi ANSI x3.9-1978 :American National Standard – Programming Language FORTRAN.” *American National Standards Institute*, http://www.fortran.com/fortran/F77_std/rjcnf.html (Dec. 10 2013)

Dolenc, M., Dolšek M. (2008). “High-throughput computing in engineering.” In: Papadrakakis, M. (ed.), Topping, B.H.V. (ed.). *Proceedings of the Sixth International Conference on Engineering Computational Technology*, Athens - Greece 2-5 September 2008. Stirling: Civil-Comp Press

Fajfar, P. (1987).” EAVEK : program for elastic analysis of multistory structures.” *Institute of Structural and Earthqueake Engineering, Department of Civil Engineering Ljubljana, Edvard Kardelj University*, Ljubljana, 130 pg.

HTCondor, http://research.cs.wisc.edu/htcondor/ (Dec. 16 2013)

Kilar, V. & Fajfar, P. (1997). “Simple push-over analysis of asymmetric buildings.” *Earthquake Engineering & Structural Dynamics* 26 (2): 233–249

Klinc, R., Turk, Ž., Dolenc, M. (2009). “Engineering collaboration 2.0: requirements and expectations.” *J. inf. tech. constr*., Vol. 14,.pg. 473-488, http://www.itcon.org/2009/31 (Dec 12 2013)

Klinc, R., Turk, Ž., Dolenc, M. (2010). “ICT enabled communication in construction 2.0” *Pollack periodica*, Vol. 5, Nr. 1, pg.109-120, doi: 10.1556/Pollack.5.2010.1.8

Turk, Ž., and Cerovšek, T. (2001). “A Prototype Portal to Web Based Collaborative Engineering.” In: Amarjit Singh, *Creative Systems in Structural and Construction Engineering* – ISBN 90-5809-161-9, Brookfield: A.A. Balkema 2001, Rotterdam, pg. 247-352