CLOUD-BASED SIMULATION OF ORC SYSTEMS

Erhard Perz[1]*, Ekaterina Rodionova[1], Stefan Bergmann[1], Fatima Dargam[1]

[1]SimTech GmbH,
Graz, Austria

* Corresponding Author
E-mail: E.Perz@SimTechnology.com

ABSTRACT

With the increasing interest in ORC systems, accurate thermodynamic models of ORC systems and their integration with other systems, such as geothermal plants, biomass combustion and gasification plants, solar power plants or industrial processes, are of great value. Thermodynamic models provide insight into process details and can be used to optimize the performance of the system. However, creating such models in the traditional way requires substantial locally installed software resources. If a project is developed as part of a collaborative effort, such resources need to be replicated for each partner. Cloud-based systems are changing the work flow patterns in many areas and make collaboration easier and more efficient. This paper describes a flexible cloud-based simulation platform for ORC processes, making collaborative simulation more accessible than with traditional solutions and offering options for new ways to work on projects.

The platform presented enables users to create, configure and solve ORC process models in the cloud. All interaction with the model, from defining the model to reporting results, is done via a web browser. It is not necessary to install any software locally. Taking advantage of recent developments in browser technology, in particular HTML5, a browser-based user interface for process modeling has been developed. It can be used with the current versions of any of the major web browsers. The user interface enables users to set up and configure the process model graphically based on predefined components. It is used to specify process parameters and to display simulation results. In this way, the system represents a complete Software as a Service (SaaS) solution for ORC process design. The underlying architecture of the system is presented.

This paper presents results obtained for example ORC process, explains the capabilities of the platform, and discusses the benefits and risks of the cloud-based approach.

1. INTRODUCTION

The ability to collaborate with others online has become a vital element of the modern workplace. Cloud-based systems for collaborative work on documents have made it much easier to develop projects with contributors in distributed locations. Cloud-based systems often don't need to install any software locally. Instead, a standard web browser functions as the user interface. One of the best known of such cloud-based collaborative software systems is Google Docs, which enables users to share and to contribute to text documents, spreadsheets and presentations.

From a user's perspective, using such systems comes with key benefits, including the following:

- Since the documents are stored in the cloud, they can be accessed from different locations. For a single user this has the benefit that he/she can access the documents wherever internet access is available. In the context of collaboration, this enables collaborators to contribute to documents without the need to send the documents via E-mail.

- Since a standard web browser serves as the user interface, the system is to a large extent hardware independent, and can be used from a wide range of devices.
- It is not necessary to license and install software locally. It may be necessary to sign up for a service agreement, but typically this does not require substantial cost and is often more cost effective than the purchase of local software licenses. It also ensures that any new functionality that becomes available can be used instantaneously by all users.

The goal of the work described here was the development of platform with the characteristics described above for creating and solving ORC process models, without the requirement to install any software locally.

All interaction with the model, from defining the model to reporting results, is done via a web browser. It is not necessary to install any software locally. Taking advantage of recent developments in browser technology, in particular HTML5, a browser-based user interface has been developed.

2. MODELLING APPROACH

The modelling approach used by the cloud-based simulation platform is the same as the one taken by the commercial process modelling system IPSEpro, from which the solver core has been used for implementation. In IPSEpro, process models are created using the Process Simulation Environment (PSE). In PSE, the user sets up the process model graphically by drawing a flowsheet using components from a model library. Required data is entered directly in the flowsheet. By drawing the flowheet and entering the data, the user implicitly creates a system of algebraic equations, which is then solved by the IPSEpro’s solver core. Results are displayed graphically in the flowsheet. The modelling approach has been described in detail by Perz et al. (1995).

IPSEpro is an open framework: component equations and physical property methods are not part of the core software. Instead, application-specific information is contained in model libraries, which can be created and modified using a special Model Development Kit. In the work described, LTP_Lib, a model library specifically developed for ORC process modelling has been used. It includes a comprehensive set of component models based on working fluids used in ORCs. The library includes models for the part-load behaviour of the components, so the user can analyze the off-design characteristics of ORC plants. The application of IPSEpro for modelling ORC systems has been described in Perz and Erbes (2011). Results obtained have among others been published by Aneke et al (2011) and by Karellas et al. (2012).

By using the solver core of IPSEpro it was possible to base the implementation of the cloud-based platform on a well modelling approach. Using the same solver core and the same model library as IPSEpro ensures that the results obtained with the cloud-based platform are identical to those obtained with IPSEpro.

3. SYSTEM ARCHITECTURE

The cloud-based simulation platform has been implemented as a web application. A web application is typically defined as a client–server computer program which the client (including the user interface and client-side logic) runs in a web browser. Consequently, the components of web applications can be divided into two categories: client-side components and server-side components, and a middleware which manages the communication between client and server.

Figure 1 illustrates the overall system architecture of the cloud-based simulation platform.

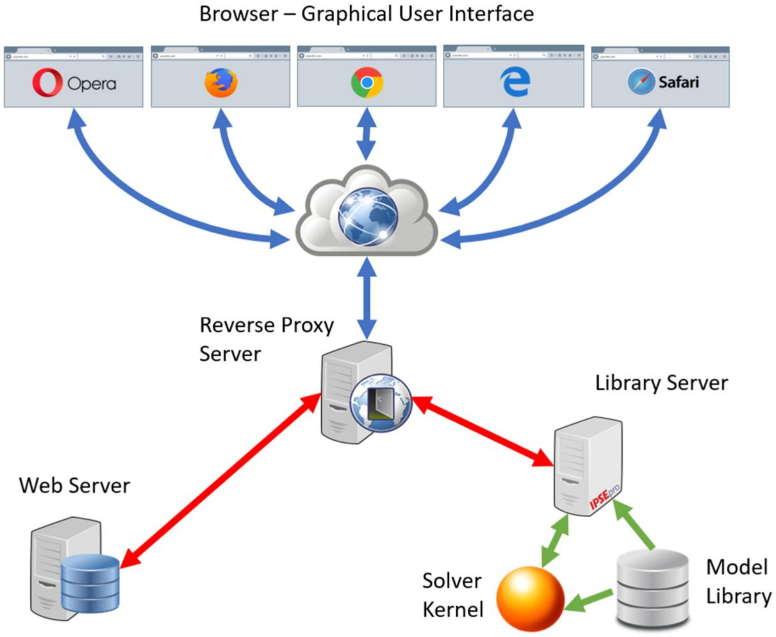

Figure 1: System Architecture

2.1 Client-Side Component
In the cloud-based simulation platform, the client side comprises two major components, a project management component and the flowsheet editor.

The project management component enables the user to log into the system, to create new projects, or select existing projects to use in the flowsheet editor. It also includes functionality to enable users to share projects with other users and to delete existing projects. For the individual user, this means that the project management component is the tool used to manage the data stored on the server.

The flowsheet editor enables the user to create and modify the flowsheet of the ORC process, to edit process parameters, to execute the system solver and to display results.

2.2 Server-Side Components
The server-side of the cloud-based simulation platform includes two major components:

- A web server with a database server, which stores all data about users and user projects. The web server also provides all parts of the interface which are not specific to a particular model library.

- A Library Server, which is responsible for providing all model-library-specific information required by the client side and which provides the capability to solve a process model that it receives.

Additionally, the server side includes a reverse proxy server which directs client requests to the appropriate back-end server. So far, the reverse proxy server, the web server and the library server have been running on the same hardware. However, in cases where the cloud-based simulation platform is subject to a heavy workload, e.g. due to a large number of simultaneous users, multiple library servers can be installed, and the reverse proxy server can be configured to act as a load balancer.

3. USER INTERFACE: FLOWSHEET EDITOR

The flowsheet editor is the central component of the cloud-based simulation platform. The flowsheet editor enables the user to create and modify the flowsheets of ORC processes, to edit process parameters, to execute the system solver and to display results.

To edit a flowsheet, a new web browser window is opened, and the flowsheet editor is loaded from the web server. Once the basic functionality of the flowsheet editor is loaded, it requests from the library server all the information about components in model library which is required to configure the user interface. This includes information about the graphical appearance of the available components (icons) as well as information about the user-accessible data in the components (i.e. variables, parameters, etc.). After this information is loaded, the user can graphically edit the flowsheet in the same way as with a desktop application.

When the flowsheet is configured, the user can trigger the solution of the flowsheet. The flowsheet editor sends the request for solving the system to the library server. When it receives the results, it automatically displays them in the flowsheet.

In order to ensure the long-term usability of the system on a wide range of platforms, it was decided to base the flowsheet editor on HTML5 (W3C, 2014). The use of HTML5 allows the system to implement dynamic web pages without the need to use any browser add-ins. Specifically, HTML5 provides a rich set of capabilities that allows the system to implement interactive graphic functionality as required in the flowsheet editor. HTML5 is now sufficiently well supported by all major web browsers (including Chrome, Edge, Firefox, Opera and Safari) and the cloud-based simulation platform has been successfully tested with all of them.

Figure 2 shows an example ORC project in Mozilla Firefox; Figure 3 shows the same project opened in Apple Safari.

4. LIBRARY SERVER

The Library Server provides two services: upon the respective request from a client, it returns the information about all component in a model library which is required to configure the user interface for using it with the respective model library. It also handles the request to solve a project.

The information about the ORC system components that are required by the flowsheet editor are extracted from the library description used by IPSEpro.

When the client requests to solve a process, it includes with the request a complete description of the structure and parameterization of the system. The library server feeds this information to the solver core, which converts it into a system of algebraic equations which it then solves. The results are then sent back to the client to be displayed graphically.

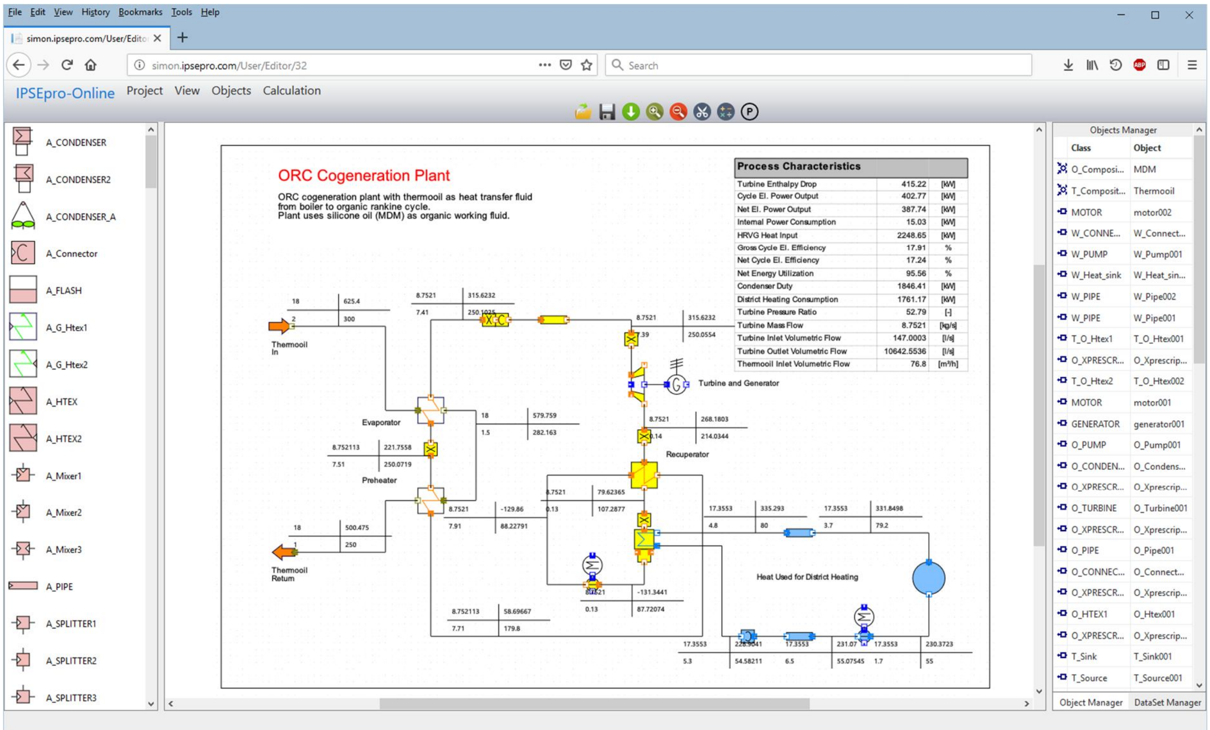

Figure 2: Browser GUI (Mozilla Firefox)

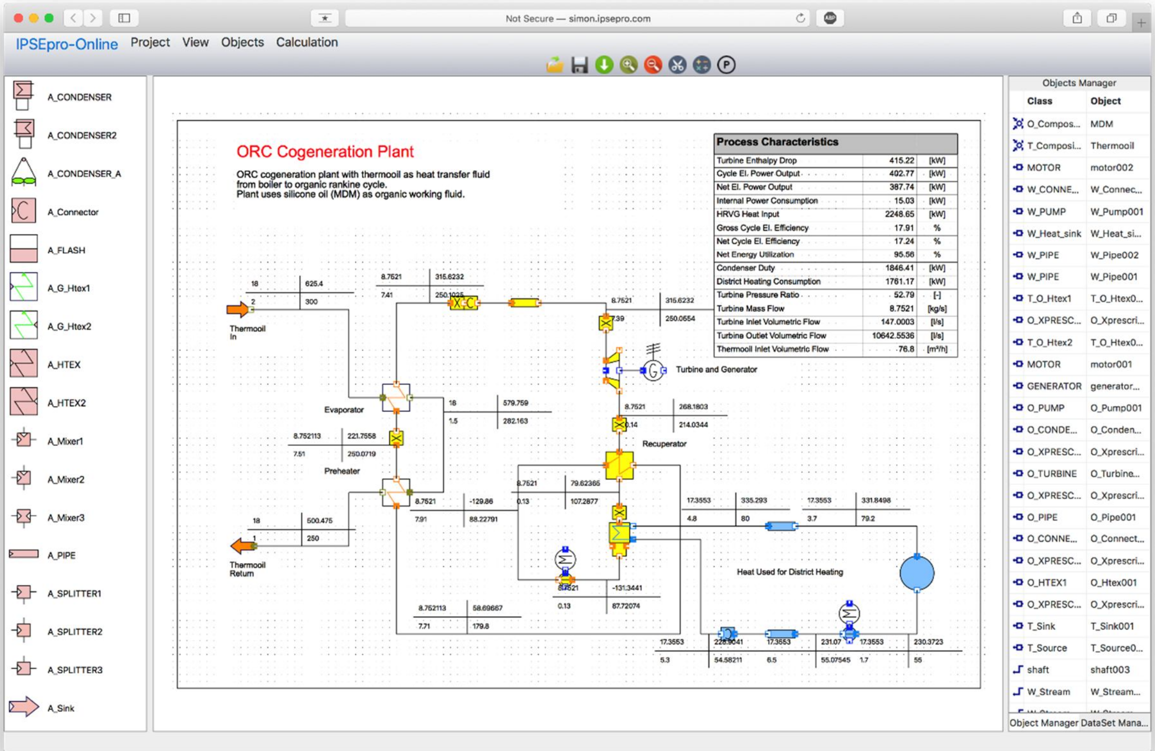

Figure 3: Browser GUI (Apple Safari)

5. PERFORMANCE CONSIDERATIONS

The usability of a cloud-based platform is, to a large extent, determined by the system performance. Slow and unresponsive interaction will inevitably reduce acceptance by users.

Several individual aspects have been evaluated during the development:
- Time for opening a project/storing projects
- Responsiveness of the user interface/browser
- Time required for solving a system

Unlike with locally installed software, the performance of a cloud-based system depends on a wide range of factors, where the speed of local hardware is only of moderate impact. Some of these factors are:
- The browser which is used on the client side
- Speed of the internet connection
- Speed of the cloud server

5.1 Time for opening a project/storing projects
The time required for opening and storing a project is determined by the speed of the internet connection. The project shown in Figure 2 takes about 3.5 seconds to open.

5.2 Responsiveness of the user interface/browser
The responsiveness of the user interface is crucial for convenient use of the platform. Comparison between different hardware and software configurations show that the choice of web browser is a major influence, due to the highly interactive nature of the user interface: Microsoft Edge shows considerably poorer performance in graphical operations than all other major browsers that have been tested. Use of Chrome, Firefox, Safari and Opera allows fluent editing of projects, while with Edge some lagging can be expected as components are moved. For other operations, e.g. entering of data, no noticeable difference between browsers was experienced.

5.3 Time required for solving a system
The overall time required to obtain a solution for calculation of a project is determined by the speed of the internet connection for transferring input data to the library server and returning results from the server back, as well as the time required by the library server to actually solve the system of equations. The solver core used by the library server is fast. The actual solution for the project shown in figures 2 and 3 takes about 0.5 seconds. Measurements show that, on average, about twice this time is required to transfer data between the client and the server, so that the overall time for solving a system is about 1.5 seconds.

6. RISKS AND BENEFITS OF A CLOUD-BASED SYSTEM

6.1 Risks and Mitigation
Using a cloud-based system comes with some risks due to the dependency on a public networked system. The system may have to cope with network failures as well as with security threads. The implementation of the cloud-based simulation platform has addressed both of these risks. It is inherent in the system architecture that a network failure will result in severe restrictions, e.g. that it is impossible to reach the library server for obtaining simulation results. However, the implemented functionality ensures that even in the case of a network failure, no data is lost. Likewise, procedures have been implemented to protect user data from unauthorized access.

6.2 Benefits

A major benefit of the cloud-based simulation platform is the fact that no software needs to be installed locally. This makes the system access easier, which is of particular benefit in collaborative research projects where a larger number of users needs to access the models. Additionally, there is no risk of version conflicts that can occur with locally installed software.

Another advantage is that process models can be readily shared, and changes are available to all authorized users virtually instantaneously.

7. CONCLUSIONS

A cloud-based simulation platform for ORC systems has been implemented and its capabilities have been demonstrated. The experience gained with the system shows that it is well suited for realistic applications and can be particularly useful for collaborative project development.

REFERENCES

Aneke, M., Agnew, B., Underwood, C., 2011, Performance Analysis of the Chena Binary Geothermal Power Plant. Applied Thermal Engineering, Elsevier, 2011

Karellas, S., Leontaritis, A.-D., Panousis, G., Bellos, E., Kakaras, E., 2012, Energetic and exergetic analysis of waste heat recovery systems in the cement industry, Proceedings of ECOS 2012 – The 25th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems, June 26-29, 2012, Perugia, Italy

Perz, E., Riesel, U., Schinagl H. A., 1995, A new approach for modelling energy systems, ASME Cogen Turbo Power Conference, August 23 - 25, Vienna/Austria

Perz, E., Erbes, M., 2011, Process Modelling of Organic Rankine Cycles, *ORC 2011, First International Seminar on ORC Power Systems, TU Delft, 22-23.9.2011.*

W3C, 2014, HTML5: A vocabulary and associated APIs for HTML and XHTML. *World Wide Web Consortium, W3C Recommendation 28 October 2014.* https://www.w3.org/TR/2014/REC-html5-20141028/, Retrieved 26 April 2019.

ACKNOWLEDGEMENT

The work presented has been partial developed within of the research project MIDES (MIcrobial DESalination for low energy drinking water) which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº685793.