20th ISPE International Conference on Concurrent Engineering
C. Bil et al. (Eds.)

doi:10.3233/978-1-61499-302-5-517

A Modelica-Based Modeling, Simulation and Knowledge Sharing Web Platform

Li Wan[a,1], Chao Wang[a], Tifan Xiong[a] and Qinghua Liu[a]
[a]*CAD Center, Huazhong University of Science & Technology, Wuhan, China, 430074*

Abstract. In order to adapt to the changes brought by mobile office, reduce the initial cost of modeling and simulation, share knowledge resources, and finally attract more customers by platform-community effect formed, the WebMWorks is presented in this paper. Deployed on cloud-computing platform, it provides user a low cost Modelica-based modeling, simulation and knowledge library sharing service of multi-domain physical system. It is based on Web, supporting multi-tenant, collaborative design, textual and visual modeling, knowledge publishing, leasing and purchasing, model commenting and communicating between users. Adopting SOA, the basic function components of WebMWorks is packed into stateless service composites, which can be deployed on cloud-computing platform and extended easily. WebMWorks is assembled with these composites, and provides service online through RIA based on browser.
Keywords. Modeling and Simulation, Knowledge Sharing, Web Platform, Cloud Computing

Introduction

The greatest stumbling block for enterprise is being short of innovation - the engine of developing, due to their lack of innovation tool or enough accumulation of knowledge. For reasons of money or time, it`s hard for enterprises, especially SMEs (Small or Middle Enterprises), to build an innovation tool and a certain scale knowledge library by themselves. Traditional modeling and simulation tools run locally, and have to download the knowledge models, which may result in a leak of knowledge. So users of traditional tools are hard to get enough knowledge models, even at a high price. Knowledge is difficult to get a greater degree of reuse and sharing.
Cloud-computing promises huge benefits for enterprises. Pay-per-use payment models, dynamic scaling of service, multi-task concurrent processing and the outsourcing of infrastructure lead to a better resource utilization and lower cost. Based on cloud-computing, software in the mode of SAAS or PAAS attracts more users and is easy to manage and maintain. It has become the new direction of software developing.
In this trend, transforming traditional innovation tool to a web application based on cloud-computing platform, providing services at a low initial cost, has become the inevitable choice of innovative tools suppliers. Meanwhile, that knowledge models are stored, modeled and simulated on cloud server, avoids leaks from knowledge sharing. By this, knowledge owner can share their knowledge for a fee without worries of leaks when it’s being reused.
Up to now, most of the web-based modeling and simulation tools are still prototype system in period of research or experiment, as presented by Eva-Lena [2] and Oscar [4]. Existing knowledge modeling tools on web, such as CADren platform [5] of Autodesk, only support knowledge modeling on specific domain in specific innovation designing period. OMweb [6], a Modelica-based modeling and simulation tool on web, only

1 Corresponding Author.

supports textual modeling. Knowledge modelers have to be spiciest of both domain knowledge and Modelica language. It's very hard to build complex system models.
Aimed to solving these problems, WebMWorks is presented in this paper. Our purpose is build a web platform deployed on cloud-computing platform, integrating Modelica-based visual modeling, simulation, knowledge sharing, community, and team collaboration tools, supporting modeling and simulation of complex physical systems, to provide enterprises innovation tools and knowledge models at a low cost.

1. Modelica, MWorks, and WebMWorks

1.1. Modelica

Modelica is a non-proprietary, object-oriented, equation based language to conveniently model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents [1]. A Modelica model is a meta-knowledge. It's saved as a text file (.mo file), similar to the class in java. From user's point of view, models are described by schematics, also called object diagrams, as shown in **Figure 1**.what a model schematic MS consists can be simply described with the follow tuple:

MS= {Attribute, Imports, Extends, Variables, Components, Annotation, Equations, Connects}.

Internally a component is defined by another schematic or on "bottom" level, by an equation based description of the model in Modelica syntax [1].

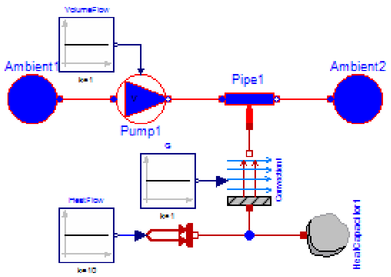

Figure 1 Schematic of Modelica Model

1.2. MWorks

MWorks is a general Modelica-based modeling and simulation platform for engineering systems which supports visual modeling, automatically translating and solving, as well as convenient post processing. The main process of MWorks is similar to the described in book of Peter Fritzson [3], as shown in **Figure 2**.

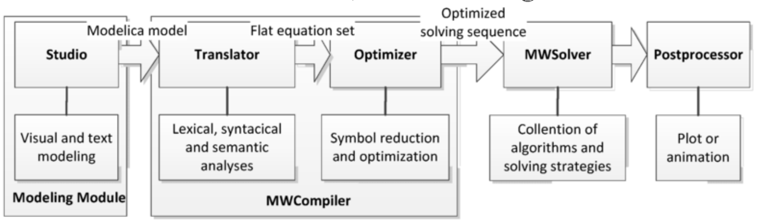
Modelica model
Flat equation set
Optimized solving sequence
Studio
Translator
Optimizer
MWSolver
Postprocessor
Visual and text modeling
Lexical, syntactical and semantic analyses
Symbol reduction and optimization
Collention of algorithms and solving strategies
Plot or animation
Modeling Module
MWCompiler

Figure 2 Main Process of MWorks

1.3. WebMWorks

As the name implies, WebMWorks is a web application similar to MWorks. Deployed on cloud-computing platform, it provides an online visual modeling, simulation and knowledge sharing service in Modelica. Through the internet, users can get the services at a low cost whenever from wherever. What's more, WebMWorks provides knowledge model sharing and team collaboration service. Users can upload their knowledge models to cloud server, and share some knowledge models with others for a fee. Since knowledge model is stored at server and both of modeling and simulation is processed at server, knowledge models is not allow being downloaded by knowledge users, which ensures that knowledge models won`t be betrayed when it`s reused. As a web platform，WebMWorks support multi-user modeling and simulation at the same time. Managed by the collaboration service based on model, a team can work together to model and simulate complex physical system.

2. Analysis

Compared to MWorks, WebMWorks Platform should have some different features, as shown in **Table 1**.

Table 1 the Compare between MWorks and WebMWorks

Feature	MWorks	WebMWorks
(1)Type	Desktop Application	Service-based Web Application
(2)Run Environment	Single Machine	Cloud Platform
(3)Use Environment	Local	Any Time and Any Place
(4)Available Library	Local Library	All Sharing Library on Server
(5)Approach to Library	Build/Purchase	Build/Rent/Purchase
(6)Amount of User	Single User	Multi-Tenant and Multi-User
(7)Amount of Task	Single Task	Multi-Task Concurrent Process
(8)Scale of Data	Data of single user	Mass Data from Multi-Tenant
(9)Expandability	Need Redeploy	User Unaware
(10)Views of Modeling	Text/Diagram/Icon	Text/Diagram/Icon
(11)Key Factor	Machine Performance	Net Performance

In order to achieve feature 0, 0 and 0，the most appropriate approach is to build client as a web application presented by browser. Browser is widely used, and provides a cross-platform support. Presented as a web pages software service can be accessed to through browsers on various operation system wherever and whenever possible.

Features (4) and (5) require that knowledge library must be stored on cloud server to get a greater degree of sharing and data safety. For the reason of feature (6), we need adopt a multi-tenancy data manage solution to isolate data from different organization.

On the side of server, we should packed function modules in MWorks into a stateless service modules to achieve multi-task processing concurrently, as feature (7) requires, by separating user data and request processing logic. All data generated by user's request should be stored or returned to client after request is processed to avoid lost or leak of data when services is request by different users in the meantime. To improve user experience, WebMWorks should provide asynchronous processing and load-balance method for service modules which may spend a lot of time or with a massive calculation.

WebMWorks have to integrate distribute file system and database to store huge amount of data of users ,pointed by feature (8), and provide a general data access interface for upper service modules.

We adopt SOA to reuse modules and ensure a better expandability of WebMWorks platform by wrap each of the function modules as a service. Changes and expansion can be deployed unwarily as Feature (9) describes.

As feature (11) implies, the key factor of WebMWorks performance is the network status. Visual modeling on web, present in feature (10) is the core function of WebMWorks. Limited to network status, it is hard to get model information immediately when network is bad. We must reduce data exchange of modeling between client and server and solve how to modeling in a bad network status when implementing WebMWorks. We can eliminate the influence of network status and improve user experience in following directions:

A. Reduce the data exchange when separating the platform into client and server, such as packing visual modeling graphic process module, which has a lot of interaction with user, into client page;
B. Avoiding transferring big data as possible;
C. Reduce data size and times of communication between client and server by caching data on client and compressing data before sent.

3. Architecture

According to the preceding analysis, The architecture of WebMWorks is designed, as shown in **Figure 3**.

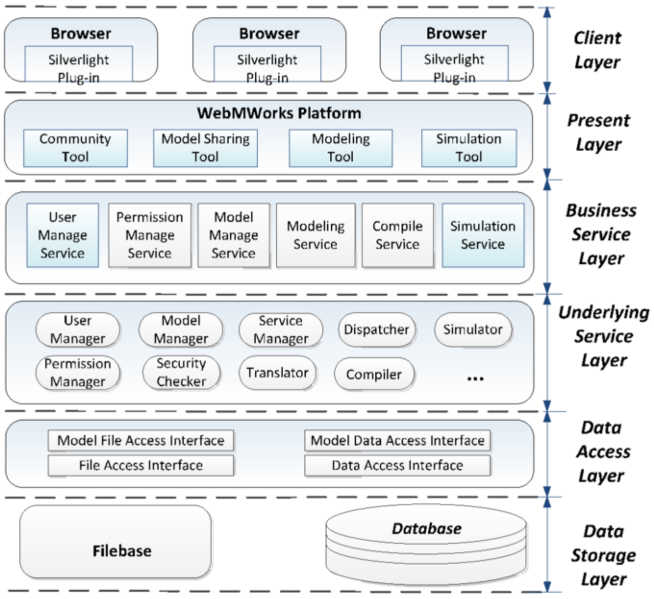

Figure 3 Logic Architecture of Platform

The Logic Architecture of WebMWorks adopt SOA Architecture, which is easy to change and deploy on Cloud-computing Platform. The Architecture is divided into 6

layers: (1)Client Layer deals with user input and provides the web interface to the application ; (2)Present Layer is the entry layer of web page and web service, user can access WebMWorks platform through different service entrance in present layer;(3)Business Service Layer provides process service for user request, including user manage service, permission manage service, model manage service, modeling service, compile service, simulation manage service; (4)Underlying Service Layer provides basic task processing service for Business Service Layer, such as translate, compile, and simulate; (5)Data Access Layer provides a general file and data access interface, especially implement a access module for model file and data; (6) Data Storage Layer includes distribute database and file base, which is the base infrastructure of platform.
Supported by general data access interface, WebMWorks can provide user manage, permission manage, models manage, modeling, compile and simulation services, which is stateless, to multi-user at the same time, no matter where and when a user requests. For task need massive calculation like compile and simulation, WebMWorks provides an asynchronous and concurrent process method to ensure that user get a better experience, as shown in **Figure 4**.

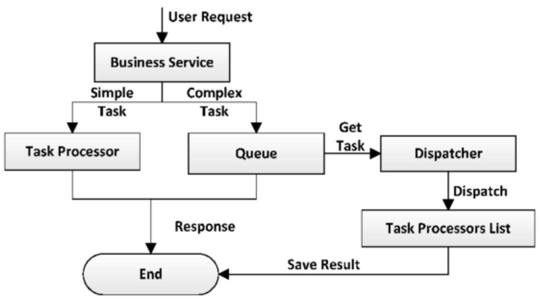

Figure 4 Synchronous and Asynchronous Task Processing

User request send to platform is packed into task by business service. These tasks is dispatched according its complexity (calculate quantities and average processing time). Simple tasks are respond by task processor immediately, while complex tasks are added into task queue waiting for asynchronous processing. A task processor can be any service module in underlying layer. It's the final processor of user request.

4. Implement and Key Techniques

We implemented the WebMWorks Platform in .Net Framework. Adopting SOA, All modules of the platform is wrapped as a service with WCF. The following section gives an overview of our major work and key techniques.

4.1. Service Encapsulation

We use WCF (Windows Communicate Foundation) to implement communication and service encapsulation. WCF is Microsoft's unified programming model for building service-oriented applications. It enables developers to build secure, reliable, transacted solutions that integrate across platforms and interoperate with existing investments [12]. The structure of service module encapsulated with WCF is shown in the **Figure 5**.

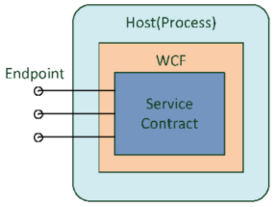

Figure 5 WCF-based Service Encapsulation

A service node consists 3 parts: host process, service contract, and endpoints. Service contract defines service-level settings, such as the namespace of the service. It`s defined by creating an interface and applying the ServiceContractAttribute attribute, the WCF service contract flag, to the interface. The actual service code results by implementing the interface. Endpoint comprises a location (an address) that defines where and how a message should be sent. A WCF service is exposed to the world as a collection of endpoints. A host is the process where service is hosted in, usually is an application that controls the lifetime of the service.

4.2. Visual Modeling on Web

The modeling page is developed in a MVVM [9] architecture using the Silverlight technology. The vector graphics and asynchronous communication of Silverlight make it easy to create interactive graphical application in the browser. While a large number of graphical operations are transferred from the server to the client, the burden of server is lightened.
We establish a mapping between an expanded svg file and knowledge model to reduce size of data exchanged. On the client side, the model is presented as an icon tag described in an expanded svg, which includes icon, attributes and parameters of model, User can build a new model schematic (an xml document) with it. On the server side, we translate model schematic into Modelica code file, which is the final file of a model, and we can generate any information of this model from its Modelica code file, including its svg file. With mapping like this, user can use models shared on server to build new knowledge model and simulate. But he can`t get the detail of knowledge model. We can even generate icon and variables of completed model, and store them in database in advance to improve access speed. Processing of visual modeling is shown in **Figure 6**

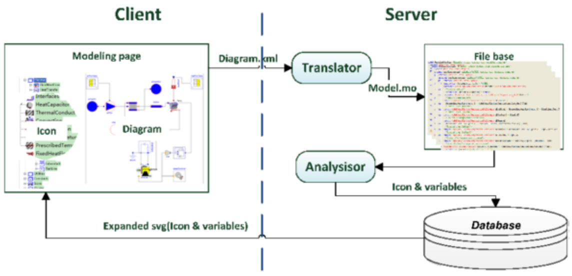

Figure 6 Visual Modeling Processing

When user is modeling, the svg tags of models on model tree are cached in client storage. User mostly needn't get any information from server while modeling, except import a new model library. If completes a model, he can upload it to server, there it will be translated into a Modelica code file to compile and simulate. User also can download Modelica code file of an available model to study or research.

4.3. Task Concurrent Processing

Facing to a large number of user requests, WebMWorks should process user's requests concurrently. In order to share the load and achieve a dynamic load balancing, we have adopted a three-tier distribution mechanism, as shown in **Figure 7**. First, user requests are dispatched among different business services. If it`s a simple task request, such as a query of data in database, it is responded immediately; otherwise the request is added to a queue as a task. When the dispatcher of a queue gets the task, including user request, it is dynamically dispatched among different underlying service node servers according to the load of the server. After receiving a task, underlying service node dispatches it again among actual processors running on the server. By this way, the large numbers of requests can be processed concurrently by different task processor.

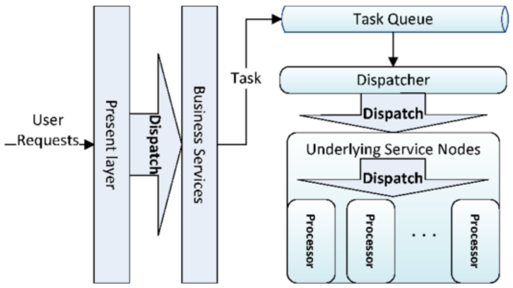

Figure 7 Task Concurrent Processing

4.4. Multi-Tenant Data Management

As a web platform, there are huge amounts of data from users. To ensure that data from different users is independent, we must adopt a multi-tenancy data manage solution. There are three approaches to implementing multi-tenancy database[10]: (1) Share Machine, (2) Share Database, (3) Share table. Since knowledge models in WebMWorks are modeling in Modelica and have the same data schema, we can implement Multi-tenant database by sharing table. Sharing table is the most suitable multi-tenancy solution for application used by a large number of small organizations, because of its better performance and higher resources utilization.

Table 2 Example of Multi-tenant Table Scheme

RowId(PK)	TenantId(FK)	ModelName	…	Val0	…	Val9
1	3	MultiBody.Forces.WorldForce	…		…	
2	2	Analog.Ideal.IdealThyristor	...		…	
…	…	…	…		…	

Sharing table means that data from different users is stored in the same tables, as illustrated in **Table 2**. "TenantId" column is added to each table to identify the owner of each row. Every application query is expected to specify a single value for this column. To allow customers to extend the base schema, each table is given several additional generic columns. These columns are of type VARCHAR. The data for the n-th new column of a table for each customer is placed in the n-th generic column of the appropriate type, after performing any necessary type conversions.

5. Case of Modeling and Simulation

Now let`s take a test. Before modeling, user must select libraries that may be imported in this model. Go to modeling page, drag icon tags on model tree to diagram area, connect them, and set value of parameters, then we can get a model diagram similar to that shown in Figure 8.

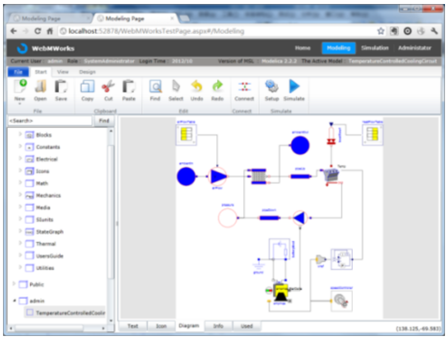

Figure 8 Modeling Page

Click "Simulate" button to start simulation, after checking and compiling, the simulation page is loaded. Select a variable in variable tree, and we will see the result plot of the variable, just like Figure 9.

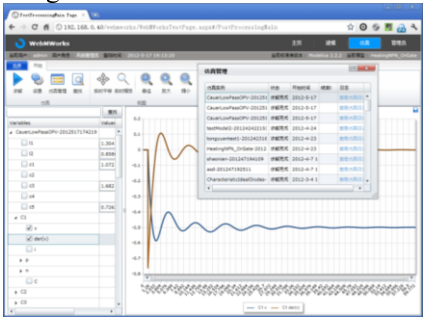

Figure 9 Simulation Page

We can summarize our platform as follow. It achieves most of our purposes except collaboration tool. When upload or download a model, user may have to wait for a while, depending on its size. Moreover, we should continue to strengthen security.

6. Conclusion

This paper presents a Modelica-based visual modeling, simulation, and sharing web platform of multi-domain physical system-WebMWorks. Deployed on cloud, it can provide users online services at a low cost, and process user requests concurrently. All users, such as SMEs, colleges, and academies, can get the service anytime from anywhere possible. By sharing knowledge on the platform, knowledge owners can benefit from reusing of knowledge models, while users of knowledge can use lots of knowledge models at a low price.

Compared to other web modeling and simulation platform, WebMWorks supports multi-domain modeling and simulation in Modelica. What`s more, it provides a visual modeling service by establishing a mapping of icon tags and models and allows team collaboration based on models, which makes modeling and simulation of huge and complex physical system easier.

However, our work is just at the beginning. We should perfect it in the future at the following aspects:

A. Optimal data exchanging and caching strategy, such as compress data before sent;
B. Improve the security manage capability;
C. Implement Account management;
D. Integrate workflow and collaborative design modules;
E. Integrate 3D visualization of simulation result.

References

[1] Modelica Association. Modelica-A Unified Object-Oriented Language for Physical Systems Modeling Language Specification Version 3.2. http://www.modelica.org , 2010.3.24

[2] Eva-Lena Lengquist Sandelin, Susanna Monemar, etc. DrModelica A Web-Based Teaching Environment for Modelica. PELAB, Programming Environment Laboratory Department of Computer and Information Science.

[3] Peter Fritzson. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley-IEEE Press, 2003.

[4] Oscar Duarte. UN-VirtualLab: A web simulation environment of OpenModelica models for educational purposes. Universidad Nacional de Colombia, Department of Electrical and Electronics Engineering.

[5] http://www.cadren.com

[6] Mohsen Torabzadeh-Tari, Zoheb Muhammed Hossain, Peter Fritzson, Thomas Richter. OMWeb – Virtual Web-based Remote Laboratory for Modelica in Engineering Courses. Proceedings 8th Modelica Conference, Dresden, Germany, March 20-22, 2011

[7] FAN-LI Zhou, LI-PING Chen, YI-ZHONG Wu，etc. MWorks: a Modern IDE for Modeling and Simulation of Multi-domain Physical Systems Based on Modelica. Modelica Association, Modelica 2006, September 4th-5th

[8] Peter Fritzson. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley-IEEE Press, 2003.

[9] http://en.wikipedia.org/wiki/Model_View_ViewModel

[10] Dean Jacobs, Stefan Aulbach. Ruminations on multi-tenant databases. Database system in Business, Technology on Web, BTW 2007 - 12th Fachtagung des GI-Fachbereichs "Database on Information System" (DBIS), Proceedings, 514-521.

[11] Stefan Aulbach, Torsten Grust, Dean Jacobs etc. Multi-Tenant Databases for Software as a Service: Schema-Mapping Techniques. Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 1195-1206. ACM New York, 2008.

[12] MSDN. Windows Communication Foundation. http://msdn.microsoft.com/zh-cn/library/dd456779.aspx