Web-based Object Oriented Modelling and Simulation using MathML

Rodolphe Zerry*†; Boris Gauss‡; Leon Urbas‡; Günter Wozny†

† Institute of Process and Plant Technology, Technische Universität Berlin, Sekr. KWT 9, Straße des 17. Juni 135, 10623 Berlin, Germany
‡ Center of Human-Machine Systems, Technische Universität Berlin, Sekr. J2-2, Jebensstr 1, 10623 Berlin, Germany

Abstract
In the world of computer aided process modelling, the potential of web-based technologies has still not been fully exploited. In particular in the scope of collaborative model development, the use of those technologies helps to solve some common problems. In this paper a new modelling environment based on a combination of MathML, XML and JAVA™ is presented that especially addresses the problems faced when it comes to sharing models as well for documentation as for execution purposes. The application is currently used in e-learning scenarios, in which students learn to model distillation column systems with the only use of their web-browser by drag and dropping human readable equations into subsystems of a given process.

Keywords: object oriented modelling, simulation, e-learning

1. Introduction
In an effort to open educational material to the world wide web and to provide more attractive learning material for students, in the last years many browser based simulations have emerged from universities and educational institutions in the field of chemical engineering (Fishwick, 2002). Besides the ability to write stand-alone client-side simulation programs (Zerry, 2002) the emerging web technologies also offer a new way of doing simulations using client-server mechanisms (Urbas, 1997). Nevertheless, up to now most of the work in web-based simulations has been narrowly focussed on building front-ends to parameterize and execute hard coded models.

At the TU Berlin, the evaluation of the students’ requirements (Gauss et al., 2001) showed that there is a need for more interaction with the provided learning material to switch from the role as passive observer to an active user state. To close that gap, we implemented MOSAIC (Modelling, Simulation, Application and Interaction for Chemical processes) which is a web-based modelling and simulation environment, specialised in writing and testing chemical process models. Our focus lied on the integration of the following three layers:

- **Specification**: To fit the modelling depth to the given engineering problem
- **Presentation**: To feed-back the selected models to the user
- **Simulation**: To use the model to solve the given problem

The paper is organized as follows: Section 2 provides a justification of the need and advantages of a web-based modelling tool and particularly how MathML and XML are

* Author to whom correspondence should be addressed: rodolphe.zerry@tu-berlin.de
used to merge the modelling and the documentation process and how JAVA™ helps to
distribute the models. The modelling language is presented more in detail within section
3. In section 4 we explain how we integrated the environment into our eLearning
structure to teach the modelling of a distillation column. Section 5 contains a
concluding summary.

2. What are the Needs and Advantage of a Web-based Modelling Tool

Today, integrated model development and simulation environments like gProms (PSE,
2003), Aspen Custom modeller (aspentech, 2002) or ModKit (Bogusch, 2001), that
support users in applying a modelling methodology, are widely used in the academical
and commercial sectors. Jensen and Gani (1998) give a good summary on the state of
the art of computer aided modelling tools and the involved technologies.

Most of the existing software is based on well structured object oriented modelling
languages, that offer the ability to define reusable elementary or composite variables
types and building blocks. To achieve this goal, the languages use object oriented
principles like inheritance hierarchies and separation of concern (Hürsch and Lopez,
1995). While most of the tools offer a comfortable way to build models for chemical
processes, only a few offer support for the documentation task and if they do so, it is
still necessary to reduce the complexity of this task (Bogusch, 2001). Another weak
point of the existing computer aided modelling tools concerns the distribution of the
models for collaborative model development and widespread execution purposes.

To examine the advantages and needs of a web-based modelling tool we have to
consider the modelling methodology on the one hand and each step of the modelling
methodology on the other hand. Web-technologies do not have the potential to alter the
modelling methodology. However they offer the possibility to change the ways of
model sharing for both execution and presentation purposes. A key aspect to the approach is the role of JAVA™, because new classes can be dynamically loaded as the program is running, thus enabling dynamic extensibility. Web-technologies and especially the use of MathML, which is the standard used for mathematical representation in the web, have also the potential to affect the way model documentation is developed. Until now, modellers are often faced with erroneous documentation because models and the corresponding documentation have to be developed in two separate tasks. In the use case of Fig. 1 a modeller from the dbta department develops an equation. MOSAIC translates the equation into JAVA™. Since the equation has been developed in MathML, the user from the zmms can see the equation in a readable format and dynamically include the JAVA™ equation into his model.

3. Presentation of the Information Layer

The application is build on an modular information framework composed of five different specification levels designed to model chemical processes. The first layer concerns the syntax for variable definition. The second layer deals with the equation definition. In a collaborative environment, it seems natural to define the equations in MathML because this standard integrates content and presentation layer. The third level refers to the unit definition, where the interfaces to other units and the use of equation are defined. The fourth level concerns the process itself in which units of the process-flowsheet are connected. The last level of this framework deals with the simulation definition. For the last three levels, the natural intersection of the web and the modelling can be found in XML (see Fig. 2). Using XSLT, a common mechanism to transform XML documents, the XML documents can be converted into a human readable format (graphics, tables...). MathML can be rendered by most of the browsers. Thus, only a web-browser is required to develop, view and browse models.

![MOSAIC language framework](image)

*Fig. 2 MOSAIC language framework*
3.1 Variable names
Currently, in most of the available computer aided modelling tools, the variables used in a unit have to be defined in the corresponding model building block. This offers the advantages that the modellers freely choose their variable names. However, especially in a cooperative workspace, modellers have to define an overhead of specifications to connect different subsystems. Therefore, we chose to use a fixed naming convention. A temperature for example has to be called T. This makes it possible to reuse equations defined for one unit in another unit without defining aliases, and when it comes to combine different subsystems to one bigger system, the connecting overhead is reduced. Since one variable type can be found more then once in a given subsystem or equation, the variables need to be indexed. For this purpose variables of subsystems and equations are grouped into pools called connectors. One variable type (for example temperature) can only occur once in a given connector. Connectors can be connected to one another.

3.2 Equations
The language in which equations are defined is content MathML. Additionally to that specification, the variables need to satisfy the variable syntax defined in the previous section. Once an equation is defined, MOSAIC generates the analytic derivatives of the equation, builds a corresponding JAVA class and compiles it. Since new JAVA classes can be dynamically loaded, the new equation is directly made available to the MOSAIC users for execution and presentation purposes. This allows them to be used in distributed environments.

3.3 Units
Each unit or subsystem is defined in a separate XML document. Fig. 3 shows an abstract of a reboiler specification. First, the user specifies what connectors or variable pools the unit is made of using the connector element. For a reboiler one could define a connector for the feed stream and for the output stream. Then using the equation element the user defines what equations can be used by the units. The connect element defines how the variables of the equation are managed. With the hypothesis element it is possible to define a set of simplifications corresponding to common modelling hypotheses.

```xml
<unit type="reboiler">
  <connector name="c1" description="Feed"/>
  <equation name="Mass balance">
    class="de.dbta.model.basic.Reboiler_MB">
      <connect equationConnector="c3" unitConnector="c1"/>
    </equation>
  </equation>
  <hypothesis name="perfect mixing">
    <simplify type="identity" source="c2:T" target="c3:T"/>
  </hypothesis>
</unit>
```

Fig 3 Unit definition in MOSAIC

3.4 Processes
Each process is defined in a separate XML document. Fig. 4 shows an abstract of a distillation column definition. Using the unit element, the user has to define what subsystems the process is made of and for each of this units using the equation element,
what equations of the possible equation set defined on the unit level are needed for the model. The subsystems are connected using the connect element.

```xml
<process>
  <unit id="1" type="reboiler">
    <equation name="Mass balance" />
    <equation name="Energy balance" />
    <connect connectorName="c1" sourceUnit="2" sourceConnectorName="c7"/>
  </unit>
  <unit id="2" type="tray">
    <!-- More units here -->
  </unit>
</process>
```

Fig 4 Process definition in MOSAIC

3.5 Simulation

Each simulation is defined by a separate XML document where the values for the chemical data is defined, variables are initialised, and the monitored variables and design-variables are chosen.

4. Teaching Modelling of Distillation Columns

To satisfy our students’ demand for a “hands on” approach to model building we developed a set of web-based modelling scenarios in which process parameters need to be calculated. For a given distillation column system, our students first have to click on an appropriate set of subsystems from a fixed set of possibilities. On the next level, for each subsystem, they choose the desired equations from a list. As stated before, this equations are displayed in a familiar manner. They then check the degree of freedom. If the searched process parameters can be determined, a cost function determining the complexity of the model depending on the amount of used equations and required physical properties, is called and the result fed back to the users. This value needs to be minimized. Table 1 illustrates some modelling possibilities for a distillation column. Once all that checks succeeded, the model is compiled and simulations can be started.

<table>
<thead>
<tr>
<th>Subsystems</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distillation column</td>
<td>Component balances, Energy balances</td>
</tr>
<tr>
<td>Rectifying and stripping sections</td>
<td>Pinch-point determination, Component balances</td>
</tr>
<tr>
<td></td>
<td>Energy balances</td>
</tr>
<tr>
<td>Each stage</td>
<td>Component balances, Overflow: Equimolar overflow or heat balance</td>
</tr>
<tr>
<td></td>
<td>Summ relations, Phase equilibrium: constant relative volatility or Free Excess-enthalpy model (NRTL)</td>
</tr>
</tbody>
</table>

Since the graphical user interface of MOSAIC is based on ProperEduct, a web-based process control system, that is also used as navigation component for an online steady state modelling lesson (Urbas et al., 2003), students can learn modelling and apply their knowledge in the same environment. To solve the equation system, the Newton-Raphson method combined with a sparse matrix solver is used. For dynamic problems, the implicit Euler method is used. The results of the modelling process for a two pressure distillation column for separation of methanol-water mixture and a comparison to the corresponding gProms model will also be presented in the lecture.
5. Conclusion

In this paper a new modelling and simulation software based on MathML, XML and JAVA™ has been presented. Currently, this modelling software is used by our students to satisfy their need for more practical experience. The environment has been integrated in an online course for steady state modelling of chemical processes. We showed that the framework is suitable to model distillation columns in different levels of detail. While the presented environment has initially been developed for educational purpose, there is clear evidence that the use of web-technologies offers new possibilities for the modelling task in collaborative and distributed environments, as it supports model sharing both on the presentation and communication layer and for execution purposes.

Acknowledgements

This work is sponsored by BMBF in the Programme “New Media in Education”.

References

Bogusch, R., 2001, Eine Softwareumgebung für die rechnergestützte Modellierung verfahrenstechnischer Prozesse. VDI-Verlag Düsseldorf. Fortschritt-Berichte VDI, Reihe 3, Nr. 705