METAMODELING – A NOVEL APPROACH FOR
PHENOMENA-ORIENTED MODEL GENERATION

Andreas A. Linninger
Laboratory for Product and Process Design
Department of Chemical Engineering, University of Illinois at Chicago
e-mail: linninge@uic.edu

Abstract
This article discusses the phenomena-oriented modeling (POM) paradigm of TechTool, a computer-aided environment for large scale chemical process models. POM is a computer-based methodology for the automatic generation of mathematical representations from physical and chemical phenomena of relevance in chemical process engineering. Interpretation of the high-level design language leads to an equivalent set of system equations. Different modeling detail and such as conceptual design, analysis of distributed systems and process dynamics require distinct modeling concepts and interpretation of the underlying phenomena. The meta-modeling feature allows engineers to define different instances of modeling languages in order to address specific modeling scope and objectives.

Keywords:
phenomena-based modeling, model generation.

1. Introduction
Several distinct approaches to computer-aided modeling and design can be distinguished. The block-oriented flowsheet simulators constitute the state-of-the-art for many applications in the chemical industry, e.g. AspenPlus, Hysys, ProSim and others. For non-standard modeling tasks in teaching as well as academic and industrial research and development, equation-oriented modeling (EOM) languages enjoy increasing popularity, e.g. Peila, 1991; Barton and Pantelides, 1993; Abacuss, 1999. These latter tools offer ample opportunities for composing process models by means of mathematical expressions. In the industrial practice, several drawbacks are often noted: (i) long development times, (ii) lack of inner organization, (iii) requirement to employ experts rather than casual users. It seems that EOM tools provide satisfactory assistance for the mathematical solution of process models; support of the computer-aided model formulation itself is limited.

Phenomena-oriented modeling (POM) is an initiative to address model generation from a system theoretical approach and aims at computer-aided support of the problem formulation. The focus lies on the specification of a phenomenological description without directly encoding detailed mathematical expressions. The main conjecture is that the concatenation of all relevant phenomena, such as reactions, material and heat transfer, etc., leads to a complete model of the system. The associated equations can be generated by an automated computer system through the appropriate interpretation of the phenomenological description. Model building by phenomena selection offers several advantages:
- The organization and composition of models is easily understood.
- Phenomena condense the information of the underlying system in highly compressed form.
- Adaptation, reuse and modifications of models can be accomplished through rearrangement and/or aggregation of phenomena rather than re-editing of the underlying constitutive equations.

The scheme of Fig. 1. illustrates the mapping between the physical system, the phenomena-based modeling elements and associated system equations. POMs are abstract descriptions of a system via high-level expressions for the physical and chemical phenomena of relevance. Models built from phenomena are associated with a set of formal mathematical expressions, i.e. system equations.

Section 2 gives an overview of prior work in modeling and simulation. From this review, we will deduce a new proposition for the organization of a computer-aided modeling architecture entitled meta-modeling. Section 3 will discuss the main features of this new approach. Section 4 highlights shortcoming of the methodology. We will close with conclusions.
3. Earlier Approaches

The graphical interface of the flowsheet simulators for the continuous petrochemical systems can be interpreted as a pictorial unit-operations centered design language. The vocabulary matches the expectations of practitioners in the petrochemical industry ideally. While the work within the scope of the model library, e.g. distillation column, offers substantial convenience, the graphical language does not support adaptation of unit models to new situations. This can only be done resorting to purely mathematical abstractions and programming. As an example take the gaseous stream containing a solid particulate phase. In traditional simulators the reader will find concepts like the particle distribution or pore structure missing and very hard to integrate.

BatchDesign-Kit (BDK) presented an operation-centered paradigm for the design of batch recipes for fine chemicals and pharmaceuticals, e.g. Linninger et al., (1998b) In BDK, batch recipes are constructed entirely through selection of operational moves. The describing system equations are generated dynamically depending on the physical and chemical properties of the mixture. No option is given to generate new operations or modify the existing vocabulary. The BDK language is therefore limited to the description of batch process.

Stephanopoulos presented the pioneering work in the area of systematic computer-aided modeling. The Model.Ia system (Stephanopoulos et al., 1990) gave the first account of a comprehensive modeling library of physico-chemical phenomena for the chemical engineering domain. The Model.Ia II design environment (Bieszczad et al., 1998) addresses the conceptual design of continuous process models in a phenomena-oriented spirit. Another advanced computer-aided modeling environment, MODKIT, was developed by Marquardt and his group (Lohmann and Marquardt, 1996; Lohmann, 1998). Both environments have graphical interfaces and can generate gPROMS code given a phenomenological description of a process model. MODKIT also offers computer-aided guidance for model evolution. It includes mechanisms for the automatic tracking of the model building phases, i.e. model history.

In all these computer-aided efforts, a fixed set of modeling elements serve as the building blocks for model generation. It is worth mentioning that phenomena-based languages have a narrower range of applicability than EOM systems due their compressed information content. More informal POM languages can therefore be applied.
for specific domain with explicit user expectations. In
general situations, a POM language may not offer
adequate language elements with a suitable mapping onto
the describing equations. In consequence, it cannot be
expected that a unique modeling language will make
phenomena-oriented modeling feasible. The multitude of
modeling environments developed in different
communities of science and engineering gives evidence of
this hypothesis.

3. Meta-Modeling

The understanding of the evolutionary character of
modeling paradigms leads us to proposition of the meta-
modeling concept. Fig. 2. gives an overview of the
TechTool architecture whose description is well beyond
the scope of this brief paper. It will allow for the gradual
adaptation of the modeling paradigm as well as for the ad-
hoc invention of entirely "new" concepts. Interpretation of
the modeling elements leads to equivalent mathematical
expressions at the highest level of abstraction. In this new
modeling philosophy, process models will be represented
by the following features: (i) a vocabulary of phenomena
that represents the building blocks of the modeling
language. (ii) An analogous mathematical representation
that capture the model behavior in terms of differential
and algebraic equations. (iii) A model paradigm that
embeds the modeling concepts, the information for their
consistent interpretation, as well as a set of specific rules
that enforce syntactic and semantic constraints of the
language.

3.1. Modeling Elements – The vocabulary of the POM
language:

Modeling elements constitute the words in the formal
POM language and may be specialized by attributes or
parameters. A conceptual language for chemical
engineering composed of two classes of modeling
elements was proposed in Linninger et al., 1998a: It
follows Marquardt’s definitions of substantial and
phenomenological complexity (Lohmann and Marquardt,
1996). Substantial modeling concepts possess a specific
physical or logical scope and usually describe separate
physical entities. Most substantial modeling elements are
characterized by extensive properties or delineated
dimensional geometric dimensions. These concepts pertain to physical
units and their connectivity, i.e. devices and links.
Phenomenological complexity stems from the behavior of
a substantial element. Phenomenological modeling
elements refer to mathematical relations that determine
the state or state transition of a substantial modeling
element. Typically they involve generic mathematical
expressions such as reactions, transport phenomena,
balance equations, and procedures for physical properties
calculations.

3.2. Model interpretation :

The model interpretation step maps each of the
modeling elements, into an equivalent mathematical
expression. In the proposed environment, procedural
agents accomplish this task. Each agent generates the
system equations for an entire class of concepts giving rise
to similar interpretation for members of same class. The
objects themselves contribute the specific part of the
information, while the agents supervise the generic code
generation. A single phenomenon may give rise to
different expressions depending on the presence of other
concepts. Context-sensitive interactions are unavoidable,
unless one wants to create a language that is as rigid as its
mathematical abstractions.

This is best illustrated by a concrete example. Consider a modeling language that recognizes kinetic
reaction models. Selection of a particular phenomenon,
e.g. homogeneous first order reaction into the context of a
reactor vessel triggers the execution of a reaction agent.
This agent automatically updates the code, i.e. equations
describing the reactor by building the specific reaction
term into the vessel by using the pseudo code of Fig. 3.
Similarly, the same agent produces the correct equations
for another object in the reaction hierarchy, e.g. second-
order reaction. In the latter case, the agent extracts the
specific reaction term from the reaction template again.

Finally, model interpretation produces equation-based
representations using the generic mathematical language
(GML) described in Linninger, 1998. The mathematical
aspects of a models can later be refined using a new object
inheritance framework as described in Linninger et al.,
1998a. In effect, all models generated using any paradigm