ON MODEL PORTABILITY

Heinz A Preisig & Tore Haug-Warberg
Department of Chemical Engineering
Norwegian University of Science and Engineering
N-7491 Trondheim, Norway
Email: Heinz.Preisig@chemeng.ntnu.no

Abstract

With the software of various commercial providers becoming mature, portability of models and associated data comes higher and higher on the agenda. CAPE-Open has accomplished a compromise getting various players on the market to communicate and agree on generating and accepting wrappers for their process models and also thermo data, both key issues in the chemical engineering's software world. Improving a step beyond wrappers requires a more basic approach, which allows to take advantage of the model structure. The design method and the used representation of the Modeller project have already proven a great degree of portability as models can be mapped into all major solver environments including Matlab, gProms, and other DAE solvers. We attempt to line out the motivation, the present and the future of this approach.

Keywords: Computer-aided, modelling, process systems engineering

Synopsis

Modelling is a core activity of any science or engineering-based profession. It is the basic means for describing the behaviour of processes, systems, activities or what ever term is being used for parts of natural processes and human-invented abstract artefacts. Having a description provides insight into the behaviour and enables various design activities that make it possible for humans to shape their world to their desire: We design and run technical systems; modify, cultivate utilize etc natural systems. Thus modelling, defined on this level, is an omnipresent activity.

Issue Portability

With modelling activities being everywhere and being so central, people's creativity has been instrumental in generating a wide and rich variety of modelling tools and instruments. However, while disciplines do grow rather independently, there is the glue of a common science base and the incentive of a common global objective that inspires people to seek cross-overs.

Portability is all about cross-over. There are a number of issues of which one should be aware of when talking about portability some of which we shall now put on the stage for closer examination:

Problem Solvers

The term MODEL is used for various types of objects such as sets of equations, instantiated equations, solved equations for a given instance etc. and SOLVERS are correspondingly defined on various levels. Most commonly one distinguishes between numerical solvers, which solve the mathematical problem given in the form of the instantiated model equations and SOLVERS that provide an entry point on a higher level, where models are assembled from model components provided in the form of libraries. Different fields have over the years generated their own solvers of this second kind (Aspen, gProms, Diva, Ascent, Modelica, Matlab toolboxes, etc etc) all of which are highly incompatible. These products are very application specific and cross-overs hardly exist. On the numerical, low-level, the most

*E-mail: Heinz.Preisig@chemeng.ntnu.no
web: http://www.chemeng.ntnu.no/ preisig/

common tools being used are linear solvers, root solvers and integrators, latter mostly in combination with the former ones as solution of differential algebraic systems often are combined with solving nonlinear implicit algebraic equations. In fact, in process engineering computations, it is recognised that root solvers use most of the computer resources.

Solvers are to one or the other degree tailored to solve a certain class of problems. The degree of tailoring, implying a constraining of the solvers application domain, is usually a question of efficiency, which may extend to a "black-white" ability of solving a particular class of problems. Thus it is solvability and efficiency that determines the choice of solvers.

Historically, the objectives of tool construction oscillates between exploiting structure to increase efficiency and width of applicability of the constructed tools. Another type of oscillatory shift is observed in how the tools are structured. Here one observes two main interacting forces, namely integration of tools and components and on the other hand segregation.

The industry built around solver technologies, to whom we refer to as PROVIDERS, have been aiming for integrated solutions, which to the user, not necessarily the designer of the tool, appears as monolithic and uniform structure with USER FRIENDLINESS being a key term implying though also information hiding.

All this does not make it easy to cross the boundaries for moving model components from one environment to another. The different fields have now generated their own environments in the form of program packages even languages, with their procedures, low-level solvers and component libraries each using their own representation of models. Part of it is motivated by the overall structure of solvers in which the problem specific instantiation sits on top of the solver implementing a particular technology, which in term makes calls to the instances of the model. The model is thus "embedded" in the solver and the models' structure match the requirements imposed by the solver.

It is thus no surprise that one observes models being almost always rewritten when they are transferred to another environment or solver.

Computing Environments

Since a lot of work is driven by the availability of technology, it is no surprise that it is often the technology that dictates the solution method and its implementation in contrast to the methodology dictating the choice of technology.

Computers, formal languages and operating systems with their respective development kids had a huge impact on the generation of a large variety in the zoo of modelling-related tools; a development, which keeps on blooming and growing with full force. Whilst this highly competitive environment has lead to a variety of computing tools and their development, it has certainly not helped in increasing the level of abstraction and thus unification of for example the user interface.

Information Data Bases

Data, mostly measured with a lot of effort and correspondingly costs, represent a significant asset to the owner. They are correspondingly well guarded and largely live a life of their own. Most of it comes carefully packaged supplied with interfaces that allow the user to query data points only thereby carefully shielding models and the underlying numerical data, "the parameters" of the fitted (semi-) empirical models.

Today, with the fairly world-wide accessibility of machines on the network, the concept of data servers has become a common way of providing access to data. This centralised solution has the obvious advantage of maintaining control over the data and its models in terms of security and maintenance, but has also the obvious disadvantage of having to handle large and ever increasing number of users and access requests.

The CAPE Community's Reaction

CAPE stands for COMPUTER-AIDED PROCESS ENGINEERING, an acronym that does not reflect the fact that this organisation is part of the chemical engineering community. This CAPE body has probably a slight advantage over other similar bodies when it goes to reactive systems as it is part of a chemical engineer's basic education. It also has a couple of other strengths, such as large-scale

simulation and optimisation, though equally it has weaknesses, in particular the systems aspect and computer-science aspect has been weakened over the past two decades.

Due to the intensive exposure to different materials and their interaction, this group was very early exposed to problem of data portability and data handling. There have thus been a number of efforts aiming at improving communications between different subgroups, businesses and fields of activities, though commercial interests have largely defeated these efforts. Beyond others, the US National Science Foundation funded efforts to achieve such goals.

In Europe it was the CAPE-Open project funded mostly by the European Community that had the most significant impact. It has so far been the most successful attempt to achieve portability involving at least the major players in the flowsheeting business such as ASPEN, PSE etc.

CAPE-Open went the only possible path in that it tried to find a compromise matching the players in contrast to proposing a complete restructuring of the tools. As a result CAPE-Open was not a revolution, but did achieve to overcome some of the commercial protectionism. There are probably two major achievements to be recognised: 1) a generic wrapper for process models 2) a generic wrapper for process-relevant data, particularly thermo data. These wrappers have not only been established, but what is most important, they have also been accepted by the provider industry and are increasingly getting used.

As mentioned, such a result could only be achieved because a compromise could be found. It also meant that the limitations are intrinsic and due to the history of the involved products, products which carry the marks of earlier limitations such as memory limitations in the 60ties and 70ties on execution and data, in the 80ties mostly on execution and speed and in the 90ties speed and complexity.

One Step Back - Two Forwards

CAPE-Open sets standards for interfaces and thus wrappers. The only way to improve things further is to exploit the structure of the models and also the associated data. The MODELLER project of this group with T Y Lee, A Mehrabani and M R Westerweele ([**?**, **?**, **?**]) constructing in a sequence a tool, which aids in synthesizing and maintaining models, is based on a canonical description built on the foundation of science, which is looked at through the glasses of system theory casting the modelling framework into an even higher level of abstraction. The results are reported in a sequence of papers and the mentioned PhD theses.

Key observations are:

- The conserved extensive quantities form a minimal state-space.
- The conservation laws describe the dynamics of a primitive system
- The plant's structure can be mapped into a network of primitive systems connected by the exchange of extensive quantities. The given directions in this graph are the reference coordinates for the respective flows.
- Extensive quantities can be transposed (reactions, phase changes and the like).
- Couplings between primitive systems are given by the continuity of the flow of extensive quantities and the continuity of the force field quantity driving the respective flow.
- Any quantity introduced as part of the description of the flows and transposition must consecutively be the result of a series of mappings from the basic space spanned by the conserved quantities.

Constructing models on this formulation yields models in the basic form, at least in terms of macroscopic field theory. This enables the mapping into any form as for example implemented in the various commercial solvers as Westerweele demonstrated in his realization of the Modeller tool generating code for Matlab solvers, two local solvers and Modelica, gProms to be written at this point in time. The product is well on its way to commercialisation.

Having realised this step forward, which certainly goes beyond the framework as laid out by CAPE-Open, we have come a step forward towards centralising synthesis and maintenance of process models, the next hurdle is then to be taken, namely the integration of (process) data.

Data

Currently users request individual data points form the connected data bases, which contain the necessary rules (models) and data. The databases are either local being distributed with the solver software for which it is tailored or separately as a standalone package or available through the net as the database may be residing on a server. An improvement of portability can again only be achieved if the data model with the corresponding data (parameters) including all the necessary algorithm required to compute point information can be extracted and included with the rest of the process behaviour description.

This approach, though, is only plausible if one tailors the model tightly to the application. In terms of thermo data, this clearly implies that the specifics on the materials & species involved as well as the phases and possible states must be known in advance. The (thermo) model being exported from a corresponding server will be strictly constraint to "interpolate" in this material-species-phase-state space.

Towards Self-Contained Models

Strictly speaking self-contained models are not possible, because there must be a minimal amount of information available that makes it possible to read and interpret the information containing the model description. Thus as a minimum, a kind of language must be available that is used to represent the model. The language should be minimal but rich enough to capture the information, which is the currently ongoing research in our group.

What to Represent

For the moment we shall limit our discussion to our common application domain, namely the process industry. A process model, as to the degree we have established it, consists of:

THE PHYSICAL TOPOLOGY: How the physical space is subdivided into primitive systems. The degree with which the space is subdivided is usually referred to as the GRANULARITY of the model. The process of subdividing the process into primitive systems is currently done manually and is the first step in the synthesis of a process model. We have currently too little knowledge as to automate this subdivision process, a subject which in it self would be worth an essay. For complex structures the network of simple systems and connections, which is the result of the subdivision, is organised hierarchically. On the basis is the simple directed graph of nodes and arcs. The nodes represent capacities for which the conservation laws describe the dynamic behaviour. The arcs represent the communication of transfer of extensive quantity and their model describes a transfer system with no capacity effects. They derive from the continuity conditions and event assumptions being made for transfer systems. Consequently such descriptions contain information about the nature of transporting the extensive quantity in the transfer system, which, in turn implies, that the corresponding material models are to be added.

The conservation laws also include terms describing the transposition of extensive quantities. In a chemical-biological context this implies reactions, phase changes and population changes. Again this involves "material" models for the kinetics.

Finally, transfer and transposition introduce fields, densities, concentrations and the like, which require "material" models lining these newly introduced quantities to the base extensive quantities. To a large extent these transformation can be seen as a generalised set of equations of states.

How to Represent

For the representation of algebraic information, one requires the obvious primitives such as identifier, variables, equations and assignments, which we will not discuss here in this brief exposition. We shall try to give only a sketch and of it not even a complete one, because research is ongoing. For the topology, things have been established.

The topology is a graph, thus a set of nodes and a matrix for the connection would be one solution. Alternatively, one can give the set of nodes and a set of duples where each duple gives the sink and the target node. Assigning to each node a unique identifier, the hierarchy can for example be represented by a nested set or nested hash table (dictionary).

System specific data is then hooked on to the

identifiers. This can also include the colouring of the graph, which indicates for example what species is where in the system, or what transposition takes place. "Colours" representing species can be put into hash tables (dictionaries) and so can identifiers of equations. Equations can also be in hash tables with left-hand-side and right-hand-side being expressions, for which again standard syntax and semantics can be defined.

Generalisation

Having established such a framework, it is relatively easy to take yet another step and break out of the chemical engineering domain by adding the component of defining the conserved extensive quantities and the conservation laws. This step is relatively straightforward. Less simple is the addition of axioms that are imposed on the definition of additional equations; for example that the definition of new extensive quantities must be using a Legendre transformation. More easy to integrate is the derivation of transfer laws, which one obtains from a combination of applying the continuity conditions and event-dynamic assumptions on the transfer system. At this point in time, we have not yet decided on how to implement the axiom definition or handling. It is foreseen that a straightforward language definition is probably the first step in that all the axioms are explicitly included. Alternatively a more generic and smaller language is generated first, which then is used to introduce the axioms. In fact the two approaches differ little if there is little or no overlap in the set of axioms.

Conclusions

Portability of process descriptions can only be improved if one exploits the structure of the models and puts it on a more basic foundation. This approach does not only facilitate a tighter control over the synthesis and handling process of models, but results also in a structure suitable for mapping into any type of programmed framework.

The result is not a wrapper, but a process of generating tailored process models. If one also has the feature to choose between different solvers, it is very much thinkable that tailored solvers can be constructed efficiently and full advantage can be taken from the structure in the solver process. It is envisioned that this will have the effect of refocusing attention on the solvers and make them to exploit the structure and characteristics of the problem, and, which is probably more interesting, also checks on the made assumptions, such as validity ranges and model switching to dynamically adjust dimensionality of the problem.

Highly portable models have a number of components:

- Topology of the plant model, namely the special decomposition being applied.
- Hierarchical decomposition of the topology.
- Details on nature of the extensive quantities being conserved and where they are present in the system as well as their transposition.
- Details on kinetics of the transposition.
- Details on the transport of extensive quantities.
- Complete set of state variable transformations connecting the secondary variables with the fundamental extensive quantities, being the ones that are conserved.
- Augmented with the material properties associated with the transport systems, the transposition kinetics and the nature of the capacities.

References