PROCESS MODELLING TECHNOLOGY: A CRITICAL REVIEW OF RECENT DEVELOPMENTS

Constantinos C. Pantelides[§¶] and Zbigniew E. Urban[§]

[§]Process Systems Enterprise Ltd.
London W6 9DA
United Kingdom

and

[¶]Centre for Process Systems Engineering
Department of Chemical Engineering and Chemical Technology
Imperial College London
South Kensington Campus
London SW7 2AZ
United Kingdom

Abstract

This paper reviews some of the key developments in process modelling technology over the last decade. These include the establishment of networks involving multiple groups of model developers and model users, and the support that process modelling tools need to provide towards the reliable and efficient operation of these process model supply chains; the trend towards open software architectures and the emergence of standards for process modelling software; the emergence of hybrid process/CFD modelling and the demands it places on process modelling tools; the significant advances in the area of physical properties; and the impact of developments in computer hardware and architecture.
The paper attempts to perform a critical analysis of the successes and failures of the last decade, and to identify the main challenges for the next.

Keywords

Process modelling. Open software architectures. Computational Fluid Dynamics. Physical properties.

Introduction

Process modelling has always been an important component of process design, from the conceptual synthesis of the process flowsheet, to the detailed design of specialised processing equipment such as advanced reaction and separation devices, and the design of their control systems. Recent years have witnessed the model-based approach being extended to the design of complex products, such as batteries, fuel cells and drug delivery systems, which can themselves be viewed as miniature plants produced in very large numbers.

Inevitably, the modelling technology needed to fulfil the demands posed by such a diverse range of applications is very different from the standard steady-state flowsheeting packages that served the process industries so well in the past. Ten years ago, at the FOCAPD'94 conference, Pantelides and Britt (1995) reviewed some of

the early developments in the area of multipurpose *process modelling environments*, i.e. software tools aiming at supporting multiple activities based on a common model. This paper represents an attempt to critically review the progress achieved over the past decade, and to identify the key challenges for the next.

We start by considering how process models are used in industry, introduce the concept of model supply chains, and analyse the implications of the latter for process modelling technology. We then proceed to consider open software architectures for process modelling tools, reviewing the significant advances that have been achieved in recent years, and attempting to point out areas where further work is required.

One of the important implications of open software architectures is that it is now possible to combine process modelling software with other related technologies, such as computational fluid dynamics (CFD). We review the reasons for, and benefits from such multiscale modelling and introduce a taxonomy of hybrid process/CFD models. We also consider the impact on process modelling of developments in the two important areas of, respectively, physical properties and advanced computer hardware.

Finally, in our concluding remarks, we attempt to place all these developments in the wider context of the utilisation of process modelling technology for both off-line and on-line model-based applications.

The process model supply chain

It is common practice in academic research for a model of a particular process to be developed and implemented in its entirety by a single researcher or by a small group of researchers. More often than not, it is the very same researcher(s) who subsequently use the model to study and optimise the process. It is perhaps inevitable that this view of process modelling has pervaded most academic research on software tools for process modelling, from where it has often been carried over into the commercial tools that are currently used in industrial practice.

The current trends in the development and deployment of models in industrial applications point to a rather different situation, reflecting the wide scope of these applications and the range of skills and disciplines that are required to bring them to successful conclusion. Model-based activities may involve fairly large number of people, often from different groups within the same company, or indeed different organisations. Model development rarely starts from scratch but builds on existing models, and again these may have been developed within the company or by external organisations, or may have been made available as part of the process modelling tool itself (e.g. in the form of pre-defined libraries).

In view of the cost and effort required for model development, another important consideration is that of re-usability of model components, i.e. their ability to be used in the context of several modelling applications. On the other hand, it is a rarely the case that two or more applications can make use of exactly the same component. The practical consequence of this is that additional effort needs to be invested to ensure that model components are more general than what would strictly be required by any particular application, incorporating a certain degree of flexibility, both parametric (e.g. regarding the values of model parameters) and structural (e.g. regarding the number of stages in a multistage equipment).

The above considerations lead to what may be described as a *supply chain of process models*. Each node in this network represents a model development activity carried out by one or more people. The activity receives as inputs process model components from one or more sources. It then makes use of these components to construct models of more specialised and/or complex processes, and to validate these models against experimental data and other a priori knowledge. It finally supplies these models as inputs to downstream activities.

The degree of model generality normally decreases as one proceeds along any path in this network. For example, a typical industrial research and development group will build models of advanced devices – reactors, complex heat exchangers, fuel cells etc., partly from first principles and partly from more generic model components (e.g. vessels, valves, pipes etc.) that typically form part of standard model libraries. The new components can then be incorporated within additional model libraries which can be used by engineering groups within the organisation to build "system-level" models (e.g. of a reaction system, an air-conditioning system or a power plant). These system-level models form inputs to engineering activities that incorporate them within models of even larger systems such as entire plants, buildings or power networks.

The models at each and every node along the above path will typically be used for various design and sub-system optimisation activities. In addition, the final models may be employed by a variety of users who are not modelling experts but use models to support a variety of activities such as plant operation, feedstock purchasing, and product sales.

Supporting the model supply chain paradigm

It is the authors' opinion that the key challenge for process modelling technology today is to provide the infrastructure for the reliable and efficient operation of the model supply chain described above. Albeit often used, the distinction between "component" and "system" models is of dubious validity or usefulness in this context: one node's "component" model is another's "system" model. At all nodes in the supply chain, the important requirements are essentially the same:

- The tools must support the development of correct models from lower-level components, potentially augmented with first-principles knowledge.

- The models constructed must be sufficiently general to allow re-usability.
- Each model must contain sufficient information to support its correct and efficient use as a component within a higher-level model at a downstream node.

Below, we consider each one of these requirements in more detail.

Developing correct models

Support for model development has been the traditional focus of both academic and commercial modelling tools. As reviewed by Pantelides and Britt (1995), multipurpose process modelling environments aim to facilitate this task by providing a means of constructing and maintaining models from either mathematical or physical (phenomena-based) descriptions. Over the past decade, a number of workers have pursued primarily the latter line of research. Notable contributions include those by Jensen and Gani (1996), Perkins et al. (1996), Dieterich et al. (1997), Drengstig et al. (1997), Westerweele et al. (1999)[1], Linninger and Krendl (1999), Linninger et al. (2000), and Bogusch et al. (2001).

Collectively, the work mentioned above represents a step forward towards facilitating the construction of correct models without requiring the modeller to be aware of the details either of the mathematical formulation of physical phenomena or of the subtle interactions between this formulation and the underlying solution methods (e.g. with respect to the index of mixed systems of differential and algebraic equations). To the authors' knowledge, however, the implementation or influence of these ideas on commercial modelling technology has been very limited to date.

Beyond the usual inertia associated with the adoption of most technological advances, one possible reason for this may be the still relatively limited applicability of the tools. Although Jensen and Gani (1996) and Dieterich et al. (1997) explicitly consider certain classes of spatially distributed systems, much of the emphasis has been on lumped systems. Unfortunately, many of today's difficult modelling problems (for which a physical, rather than a purely mathematical, description might be of real value) concern distributed systems, especially in 2 or more dimensions where there is an interaction between fluid mechanics and other phenomena[2]. In fact, the detailed modelling of most lumped systems of practical interest to process engineering is already addressed by the model libraries provided as standard with commercial modelling tools.

Supporting abstraction and re-usability of model components

Irrespective of the manner in which they are constructed, it is highly desirable for all models to be sufficiently general to allow a certain degree of re-usability. This aspect is already well catered for in commercial modelling tools which allow the definition of models as abstract entities encompassing not only a high degree of parametrisation, but also structural generality – for example, regarding the number and identity of the species being present in the system and the chemical reactions taking place, the number of stages in multistage components and their connectivity, and so on. On the other hand, most of the academic developments mentioned above appear to aim at the development of rather specific models for specific applications – all the way from determining the fundamental physics of a unit operation to the provision of specific values of the variables representing input and operating conditions. Unfortunately, such an approach is not directly compatible with the model supply chain paradigm described earlier in this paper.

Removing the information bottleneck

The third major requirement for supporting the model supply chain is related to the correctness and efficiency of utilisation by a supply chain node of model components produced by another node.

Incorrect or inefficient usage of a component is often caused by an *information bottleneck* existing between the component's developer and the component's user, leading to insufficient knowledge about the model's characteristics and behaviour being transferred between the two. As pointed out by Bogusch et al. (2001) in their comprehensive description of the MODKIT system, "*a model is not just equations*". In addition to the mathematical description of a system's behaviour, a model needs to include information on the underlying assumptions, the degrees of freedom and initial conditions, as well as other documentation such as the decision trail in terms of the issues raised during model development, the various alternatives ("positions") considered for the resolution of each such issue, and the arguments in favour or against these positions (Bogusch et al., 2001). The range of applicability of the model, expressed in terms of bounds or more complex inequality constraints involving its variables, may also be an important consideration.

It should be noted that there are several complications associated with this ostensibly simple information: for example, there may be multiple combinations of a model's variables, each of which corresponds to a valid degree-of-freedom or initial condition specification; some initial condition specifications may be consistent only with certain degree-of-freedom specifications; and the actual

[1] For a more detailed account of this work, see also Westerweele (2003).

[2] The formulation of correct boundary conditions for such problems requires a significant degree of mathematical understanding and skill, see Martinson (2000), Martinson and Barton (2001) and Neumann (2004).

values specified for the degrees of freedom or the initial conditions may have to obey certain constraints, e.g. lie within certain bounds, sum up to 1, be in ascending order and so on.

Perhaps an even more important part of a model is the knowledge required for initialising a model's instance. This simply means obtaining a feasible point, i.e. a set of variable values that satisfy the model's equations, often at steady-state. This rather mundane task is often important in itself and, almost always in practice, it is a pre-requisite for the more sophisticated model-based applications (e.g. steady-state simulation of flowsheets involving this model instance; dynamic simulation; steady-state and dynamic optimisation). However, in our experience, obtaining such a feasible point still represents one of the most important obstacles to the efficient utilisation of modelling technology. Mathematically, this is a result of the difficulty of obtaining a solution of a large set of nonlinear algebraic equations, often starting from a set of bad initial guesses.

There have been several attempts over the past decade to devise improved numerical methods for solving these problems in the context of process modelling applications. These have included homotopy-continuation methods that take account of the bounded nature of the variables in process models and the intrinsic sparsity of the equations (Paloschi, 1995, 1997, 1998a), and methods based on interval arithmetic (Schnepper and Stadtherr, 1996, Gau and Stadtherr, 2002). However, the application of these techniques to general process modelling problems appears to have been limited to date[3].

An alternative to generic mathematical solution algorithms is to employ methods tailored to initialise particular types of model. The essence of the approach is to devise and solve a sequence of problems of increasing difficulty, with the solution point of one problem forming an initial guess for the solution of the next one. For example, many reactor models can be initialised by solving a 3-step sequence involving respectively (a) setting the pre-exponential Arrhenius constants to zero, thereby solving a much easier non-reacting system; (b) restoring the Arrhenius constants to their correct values, but setting the enthalpies of reaction to zero, thereby solving an isothermal (or near-isothermal) problem; and (c) restoring the enthalpies of reaction to their correct values, thereby obtaining the solution of the original problem. In general, the steps in these sequences may either be implemented in a discrete manner or employ a homotopy-continuation involving the continuous evolution of one or more physical parameters (e.g. the enthalpies of reaction in the above sequence). A more sophisticated example of an initialisation procedure applied to general distillation column models has been given by Fletcher and Morton (2000).

In fact, initialisation procedures of the type described above are routinely devised by the developers of almost all non-trivial models, often as an integral part of the evolutionary process of developing the model. What is required is a formal mechanism for conveying this knowledge from the developer of the model component to its user(s) in the model supply chain. In the past, written documentation has been the main means of achieving this. However, this is unsatisfactory for various reasons: written documentation is often incomplete; even if it is complete, it quickly becomes out of date as the model is updated to correct problems or address new requirements; and, unfortunately, it is sometimes misunderstood or simply remains unread.

Three generic requirements for modern modelling tools can be formulated in order to address these problems:

- During the development of a model component, a modelling tool needs to provide formal mechanisms for the model's developers to record all relevant knowledge available to them in a complete and unambiguous manner.
- Whenever an instance of a model component is being used, a modelling tool should, to the maximum extent possible, act on this information automatically (e.g. to implement an initialisation procedure).
- If automatic action is not possible or desirable, then the modelling tool should make the information available to the model component's user in a clear and standardised manner (e.g. to request enforcement of constraints on the specified values of degrees of freedom).

Achieving the above objectives is a non-trivial task, especially in view of the fact that, as explained earlier in this paper, in the interests of re-usability, model components have to be defined in an abstract parametrised manner.

Open software architectures for process modelling

The need for open software architectures in process modelling tools has been discussed by several authors, including Pantelides and Britt (1995) and Kakhu et al. (1998). Open architectures are essential to allow software fulfilling diverse functions, originating from diverse sources and implemented in diverse computer languages to be used together to achieve complex tasks in the context of process modelling.

The CAPE-OPEN standards

In view of the above, it is fortunate that this area has witnessed some of the most significant advances over the past decade through the results of the CAPE-OPEN and Global CAPE-OPEN projects (see Braunschweig et al.,

[3] In recent years, internal arithmetic techniques have been finding an increasing number of successful applications in more restricted problem domains such as the prediction of phase equilibria.

2000). These international initiatives have adopted a decomposition of process modelling software into Process Modelling Environments (PMEs) and Process Modelling Components (PMCs). The latter are software components that fulfil a well defined and relatively narrow function, such as the computation of physical properties or chemical reaction rates, the simulation of a particular type of unit operation, and the solution of systems of equations of a particular kind. On the other hand, PMEs are the general host environments that allow the user to define a process model and to perform different model-based activities with it; to do this, they make use of PMCs and other software components.

The CAPE-OPEN initiative has proposed a set of standards for some of the most commonly used classes of PMCs. The standards for physical properties calculations and steady-state unit operation modules are now well established, and have already been implemented to varying degrees in a number of PMEs including Aspen Technology's Aspen Plus® and HYSYS® software, SimSci-Esscor's PRO/II® and Process Systems Enterprise's gPROMS®. This development has allowed, for instance, physical property software from a variety of sources to be used directly within these PMEs. It has also made it feasible for software components from one of these packages to be used within the other; it is, for example, now possible to use physical properties from Aspen Plus directly in gPROMS; and to use a complex unit operation model described in gPROMS (and possibly making use of Aspen Plus physical properties) directly within an Aspen Plus flowsheet.

Less well established CAPE-OPEN PMC standards include those for physical property databases, petroleum fractions, chemical reaction and electrolytes, numerical solvers for the solution of systems of various types of equations (linear and nonlinear algebraic, differential-algebraic, and partial-differential algebraic), and also numerical optimisation solvers for linear, nonlinear and mixed integer nonlinear programming problems. These exist at various stages of development, approval and testing.

A more complete description of the CAPE-OPEN standards can be obtained from the web-site[4] of the CAPE-OPEN Laboratories Network (CO-LaN), the non-profit organisation that maintains the standards and works towards their dissemination.

Open software architectures outlook

The developments described above have allowed well-defined software components (PMCs) to be incorporated seamlessly and with minimal effort within process modelling tools (PMEs). Increasingly, there is a need for a different and rather more sophisticated type of integration in which entire process models and model-based computations are incorporated within even higher-level applications. A major class of the latter is advanced process automation applications such as those for real-time optimisation, model-based control, abnormal situations management and other decision-support tools. Another class is that of operator training applications.

The applications mentioned above are substantial software programs in their own right. A common characteristic of all of them is that they require access to process models of varying levels of detail and mathematical complexity; they also need to be able to solve or otherwise manipulate these models in a variety of ways.

It was common practice until recently for each such software application to provide its own means for the user to define these models and to perform the necessary manipulations. However, this approach is difficult to sustain as the process models are becoming increasingly detailed and complex. Moreover, it results in an unnecessary and undesirable duplication of modelling effort between off-line modelling activities (e.g. for the purposes of process design or operational troubleshooting), and on-line activities. This duplication, and the potential for inefficiencies and inconsistencies that it entails, is becoming difficult to justify given the increasing availability of computer power and the improving reliability of process modelling technology. Finally, process modellers are becoming much more demanding regarding the sophistication of the user interfaces that they employ for model development, validation and testing. The cost of developing and maintaining these functions in many different software applications is prohibitive.

All of the above factors point towards the need for process modelling tools to be able to act as providers of model construction, maintenance, manipulation and solution services to higher-level software. A few first steps towards the realisation of this "model server" concept (Pantelides and Britt, 1995) have been made over the past few years. For example, the CAPE-OPEN standard for an Equation Set Object (ESO) allows external applications to access information on the mathematical form of a model comprising differential and algebraic equations (e.g. the number of variables and equations, the structure of the Jacobian matrix, the current values of the model variables) and to request some basic numerical calculations (e.g. the evaluation of the equation residuals and Jacobian) to be performed. The ESO concept has been implemented in gPROMS and has already found use in several industrial and academic applications in the areas of advanced model-based control and dynamic optimisation. Also, the gPROMS engine (gSERVER) provides a software interface that allows 3rd party applications to execute various types of model-based activities, interacting with them via an event-based mechanism and receiving diagnostic and other information via a standardised messaging interface.

[4] http://www.colan.org

It is already clear, however, that a level of integration that goes well beyond the basic mathematical/numerical description of a process model will be necessary for some applications. The latter need to be able to access and, where appropriate, manipulate all aspects of a process model, including, for example, the hierarchical topological structure of the model, the abstract symbolic form of the model's equations, and the recommended degree-of-freedom and initial condition specifications for a model component. External applications should also be able to construct and manipulate descriptions of complex model-based activities, such as combined continuous/discrete dynamic simulation involving non-trivial sequences of tasks, mixed integer dynamic optimisation, or model reduction for the generation of simpler models for real-time applications.

To fulfil these requirements, process modelling tools will need to embrace much more open software architectures than has hitherto been common, and to provide published interfaces to their innermost components and functions. It has to be recognised that such a development potentially entails both technical problems and, in the case of commercial tools, business risks. Nevertheless, it is the authors' experience that there are satisfactory ways of addressing these issues.

Process modelling tools for multiscale modelling

A key challenge to process modelling today is the accurate description of the interactions between, on one hand, mixing and fluid mechanics, and on the other, homogeneous and heterogeneous reaction and mass and heat transfer. It is largely these interactions that determine the performance of advanced equipment for separation and chemical reaction and the properties of the products that they produce.

Despite the increasing capability of process modelling tools to model spatially distributed systems (see, for example, Oh and Pantelides, 1996), their ability to construct and solve 2- or 3-dimensional descriptions of fluid mechanics and mixing, especially in equipment of irregular geometry, is still quite limited. On the other hand, computational fluid dynamics (CFD) tools are designed for precisely this task but have difficulty in dealing with complex reactions either because of the inherent nonlinearity of these phenomena or because of the large numbers of species that need to be tracked, or both. Similar considerations pertain to the case of processes described by population balances, complete descriptions of which would require tracking large numbers of quantities around the system[5]. Moreover, CFD tools have difficulty in resolving certain important phenomena that operate on much finer spatial scales than the size of a typical discretisation cell; for example, the performance of heterogeneous catalytic reactors is often determined by highly nonlinear phenomena of reaction, multicomponent mass transfer and heat transfer that take place on the surface and in the pores of catalyst particles of diameter well under 1 mm, and in the even thinner laminar sub-layers surrounding these particles.

Over the past decade, a number of hybrid models involving the combined application of process modelling and CFD tools have been proposed, implemented and tested. Despite their diversity, most of these models can be classified into a small number of categories, as detailed below.

Category I: Hybrid multizonal/CFD models

These are models in which the domain of interest is represented as a network of (usually, but not always, well-mixed) zones implemented within a standard process modelling tool.

This approach is appropriate in situations in which the description of the important phenomena taking place in the system would require a relatively large number of variables which also participate in convective and/or dispersive transport within the equipment. It is also useful in cases involving very nonlinear phenomena. These characteristics render such processes practically impossible to model with currently available CFD technology. Examples include reactions involving large numbers of chemical species (see, for example, Falcitelli et al., 2002), crystallisation processes described by complete distributions of particle sizes (see Urban and Liberis, 1999), and modelling of bioreactors involving distributions of cell masses (Bezzo et al., 2003). The work by Bauer and Eigenberger (1999, 2001) on modelling of bubble column reactors is distinguished by the fact that the zones are not well-mixed but are modelled as 1-dimensional distributed systems.

Multizonal (or "multicompartment") models are not particularly new. However, a key problem in their use has always been the determination of the fluxes of material and energy between adjacent zones. These can be estimated via the use of a CFD sub-model which attempts to resolve the fluid mechanical phenomena based on a much finer discretisation of the domain of interest. Since each zone of the multizonal model represents a subset of the CFD model's discretisation cells, the inter-zonal fluxes can be computed from the CFD solution.

A secondary role for the CFD sub-model is to compute fluid mechanical quantities which have a significant effect on the phenomena taking place within each zone. Examples of such quantities include the turbulent energy dissipation rate which affects the rate of heterogeneous nucleation in crystallisation (Urban and Liberis, 1999) , and the fluid strain which affects the rate of mass transfer between gas-phase and dissolved oxygen in bioreactors (Bezzo et al., 2003).

[5] For example, the detailed description of a particle size distribution in crystallisation via an appropriate discretisation could involve tens or hundreds of values.

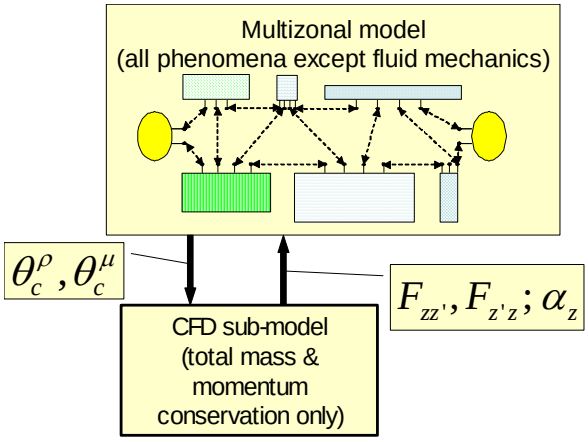

Figure 1: General methodology for hybrid multizonal/CFD modelling (Bezzo et al., 2004)

There is little doubt regarding the usefulness of hybrid multizonal/CFD models and the improved predictive accuracy that they attain. The real challenge is to formalise these models to the extent that they can routinely be formulated and solved in the context of standard process modelling tools. The recent work of Bezzo et al. (2000, 2004) represents an attempt to put in place the necessary theoretical framework. A schematic illustration of the hybrid multizonal/CFD modelling methodology is shown in figure 1. The multizonal model comprises a network of zone models, each having a number of ports that allows it to be connected to other zones via bi-directional interfaces. The mass flowrates $F_{zz'}$ and $F_{z'z}$ in each of the two directions of the interface between two zones z and z' are computed via the solution of a relatively compact CFD sub-model which comprises only the total mass balance and the momentum balance equations, in addition to parametrised relations for the fluid density and viscosity. Thus, the CFD sub-model does not attempt to track any intensive properties (e.g. temperature, composition or particle size distribution) other than fluid density and pressure. The fluid is assumed to be pseudo-homogeneous and the parameters θ_c^{ρ} and θ_c^{μ} involved in, respectively, the density and viscosity parametrisations are computed by the multizonal model[6] and may be different for each cell *c*. Additionally, the model for zone z may involve one or more fluid-mechanical quantities α_z which are, again, computed by averaging certain aspects of the solution of the CFD sub-model over the cells that correspond to the zone.

Overall, the framework described above results in a fully-coupled hybrid model that has to be solved simultaneously. In practice, experience with the systems to which this methodology has been applied to date indicates that a first-order (successive substitution) iterative scheme between the multizonal and the CFD sub-models is sufficient, converging within a few iterations provided appropriate density and viscosity parametrisations are selected.

It is worth noting that, in systems where fluid mechanics operate on a much shorter time constants than the rest of the phenomena, a steady-state CFD sub-model can be used within a dynamic multizonal model, thus allowing dynamic simulation of systems that would simply be intractable using "pure" CFD technology.

An open issue regarding all hybrid multizonal/CFD models is that of determining an appropriate partitioning of the spatial domain of interest into a relatively small number of zones. More specifically, it is important to identify zones for which the well-mixedness assumption is reasonably justified. At present, this is done in a mostly *ad hoc* fashion, by carrying out preliminary CFD calculations and inspecting the solutions obtained. Some ideas towards a more automated approach have recently been proposed by Bezzo (2002) and Bezzo and Macchietto (2004).

Category II: Models involving complex source terms

In many practical situations, the generation (or "source") terms within the species mass, momentum and energy balance equations are complex functions of the species concentrations, temperature, pressure and velocities.

A typical example is that of heterogeneous reactions taking place on the surface of catalyst particles that are suspended in a liquid. The rates of these reactions depend on the concentration of various species[7] and the temperature on the particle surface, and these, in turn, are related to the bulk compositions and temperature via the multicomponent mass transfer and heat transfer phenomena taking place in the laminar sub-layer surrounding each particle. Ultimately, the rate of generation of any species that appears in the bulk fluid is a well-defined function of bulk fluid properties; however, evaluating this function involves the solution of the Maxwell-Stefan equations of multicomponent mass and heat transfer, possibly coupled with homogeneous reactions taking place in the laminar sub-layer, and subject to boundary conditions determined by the true rate of the heterogeneous reactions on the catalyst surface.

A somewhat less obvious example of this category is that of modelling multitubular reactors used for performing exothermic catalytic reactions. Such reactors typically involve thousands of tubes packed with catalyst

[6] In the simplest case, the fluid density and the viscosity are taken to be constant within each zone, being functions of the corresponding composition and temperature as determined by the multizonal model.

[7] These may include both the species occurring in the bulk of the fluid and intermediates occurring only on the catalyst surface.

particles. The heat removal needed to avoid thermal run-away is achieved by embedding the tubes in a shell through which a cooling medium is passed. The optimal design of these reactors requires the accurate characterisation of the heat transfer within the catalyst bed, from the bed to the tube walls, and from the tube surfaces to the cooling medium. The latter contribution is significantly affected by the coolant flow within the shell and the latter's detailed design (e.g. the number and positioning of baffles). In view of the large number of tubes and their small diameter relative to that of the shell, a reasonably accurate CFD model of the shell can be constructed by considering the shell as a porous medium. In this case, the tubes need to be modelled only implicitly, acting as an energy source term which is a function of the local temperature and velocity of the surrounding coolant[8]. However, the computation of this source term involves the solution of a set of coupled partial differential and algebraic equations representing the fluid flow, reaction and heat and mass transfer phenomena taking place within each tube.

In principle, both of the examples mentioned above could be described by "pure" CFD models. However, such an undertaking may not be feasible in practice. In the case of reactors with suspended catalyst particles, CFD tools already allow for arbitrary source terms to be inserted in the form of user-defined functions. However, both the correct formulation of the single-particle equations and their robust and efficient solution are non-trivial tasks which may be more suitable for a process modelling tool. The latter can calculate the effective species and energy source terms for values of the bulk composition and temperature provided to it by the CFD tool (see figure 2). The thickness of the laminar sub-layer is a function of the turbulent energy dissipation rate, a quantity that is also computed by the main CFD model.

In the case of multitubular reactors, the explicit inclusion of tens of thousands of tubes within the CFD mesh could render the latter too large to be handled with currently available computer resources. Also, the equations describing the phenomena taking place within the tubes may be too complex and/or nonlinear to be handled by the CFD solvers. On the other hand, process modelling tools can solve the equations for a single tube with relative ease, determining the effective energy source term for a given variation of the coolant temperature surrounding the tube and the coolant-tube heat transfer coefficient.

In both cases, the hybrid model comprises a CFD model of the processing equipment coupled with a sub-model used exclusively for the computation of the effective source terms. This computation has to be very efficient as it is carried out a large number of times during a CFD calculation[9]. Fortunately, each such solution normally starts with a good initial guess (i.e. the final point of the previous solution) which is within the radius of local convergence of the quasi-Newton iterative schemes that are typically used by process modelling tools; consequently, converged points are usually reached within a few iterations. Of course, the computer implementation of such hybrid models is possible only if the software architecture of the process modelling tool allows it to be embedded efficiently within other software, along the lines discussed earlier in this paper.

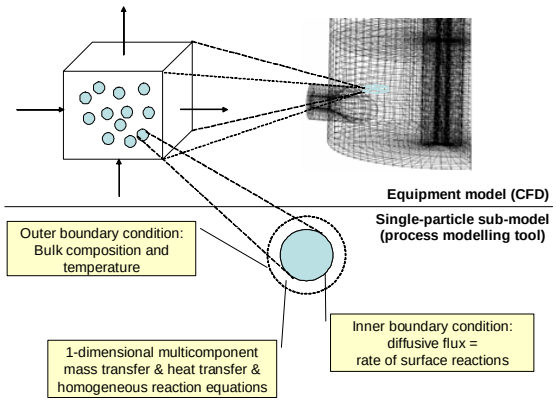

Figure 2: Multiscale modelling of reactors with suspended catalyst particles (Process Systems Enterprise, 2003)

Despite their benefits, it is important to recognise that category II hybrid models suffer from certain fundamental limitations arising from the fact that responsibility for the solution of the model ultimately rests with the CFD tool. Thus, these models are practically applicable only to systems involving a relatively small number of quantities (e.g. species concentrations) being tracked by the CFD model. Moreover, the CFD iteration may not converge if the source terms are very nonlinear functions of these quantities. Finally, the ability of these models to be used for dynamic simulation is quite restricted.

Category III: Spatial boundary-coupled hybrid models

The hybrid models of the first two categories involved CFD and process models describing the same spatial domain, but focussing on different phenomena and/or levels of abstraction. In contrast, category III models comprise sub-models which represent different spatial domains which are clearly separated by a spatial boundary. The CFD sub-model typically describes a gas or liquid-phase sub-system, often of irregular geometry, whose

[8] The (implicit) presence of the tubes also introduces a momentum source term in the form of an anisotropic resistance force exerted on the fluid.

[9] For example, the suspended catalyst particle equations may have to be solved hundreds of thousands of times during a typical CFD calculation, corresponding to the particles within every discretisation cell at each iteration of the CFD model

behaviour is dominated by fluid mechanics, although it may involve additional phenomena. On the other hand, the sub-system described by the process modelling tool is often a solid phase one which does not, of course, involve any fluid mechanics but may involve other complex phenomena.

A category III hybrid model has recently been used by Marias (2003) for the detailed modelling of a rotary kiln incinerator. This combines a 1-dimensional (plug flow) model of the solid phase in gPROMS with a 3-dimensional model of the gas phase implemented in the Fluent® CFD tool. The two sub-models are coupled by the fluxes of the products of solid phase pyrolysis from the solid to the gas phase, and, in the reverse direction, the convective and radiative heat flux arising from the gas-phase combustion of some of these products in air.

Another category III model was used by Urban et al. (2003) for modelling a solid oxide fuel cell (SOFC). This combines a 3-dimensional CFD model of the fuel and air channels, with a 1-dimensional gPROMS model of the solid membrane that separates the two channels (see figure 3). Whilst the CFD sub-model focuses on the fluid mechanics and mixing in the two channels, the solid membrane sub-model incorporates a detailed model of the O^{2-} migration in the electrolyte, the multicomponent diffusion and reactions in the anode, the heat generation at the interfaces between the various parts of the membrane, and the heat conduction throughout the system.

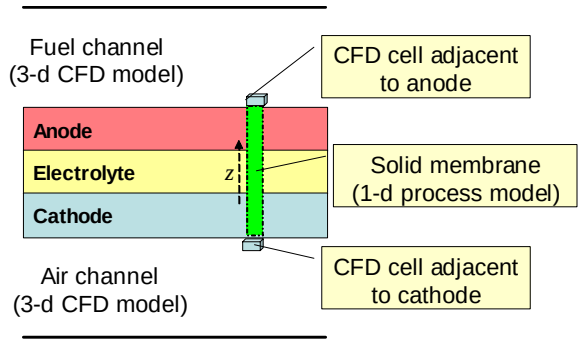

Figure 3: Category III hybrid SOFC model (Urban et al., 2003)

The two sub-models are coupled: given the compositions and temperatures in two CFD discretisation cells which are positioned adjacent to the anode and cathode respectively and opposite to each other, the process model is solved to determine the species and heat fluxes from the solid to the gas phases at these two points; these form the boundary conditions for the CFD sub-model. This calculation has to be repeated at each CFD iteration for every pair of cells adjacent to the anode and cathode surfaces.

Category III hybrid models are essentially CFD models with sophisticated boundary conditions. They are, consequently, subject to the same potential limitations that have already been outlined for the case of category II hybrid models.

Physical properties and process modelling

The role of physical properties in process modelling was considered briefly by Pantelides and Britt (1995). Some of the more recent developments and the evolving role of physical properties in process modelling have been reviewed by Gani and O'Connell (2001) and Gani and Pistikopoulos (2002).

One of the key developments in this area over the past decade has been the increasing use of equations of state (EOS) with a more fundamental physical basis than the cubic EOS and their extensions commonly used for process modelling in the past. A prime example is the EOS based on Statistical Associating Fluid Theory (SAFT, Chapman et al., 1989) and its more recent variants such as HR-SAFT (Huang and Radosz, 1990, 1991), SAFT-VR (Gil-Villegas et al., 1997), and PC-SAFT (Gross and Sadowski, 2001). These have been used with increasing success for the modelling of complex mixtures such as those involving polymers (see, for example, the recent review by Sadowski, 2004, and the work by Cheluget et al., 2002 and Bokis et al., 2002 on making use of these developments in the context of process modelling), associating mixtures with strong hydrogen bonding (Galindo et al., 1997, 2002, Gross and Sadowski, 2002) and electrolytes (Gil-Villegas et al., 2001, Patel et al., 2003). More general reviews of SAFT and its applications to different types of materials have recently been published by Müller and Gubbins (2001), Economou (2002) and Paricaud et al. (2002).

All of the above developments are highly beneficial to process modelling: a good theoretical model of material behaviour is necessary for physical properties calculations to be reliable over wide ranges of conditions. It is, however, important to recognise that having such a model is by no means sufficient. At least two additional requirements also need to be met:

- All physical property models involve parameters that need to be fitted to available experimental data. The quality of estimation of these parameters needs to be assessed not only via the usual comparisons between predicted and measured data, but also by the statistical significance (e.g. as measured by the confidence intervals) of the parameter estimates. This is particularly true of SAFT-based EOS which involve a relatively large number of parameters, especially in the case of molecules with several association sites. In fact, these confidence intervals may have a crucial effect on the uncertainty involved on any decisions (e.g. design calculations) derived using these models.
- Even if a physical properties model has a solid physical basis and sufficiently accurate estimates

of its parameters have been obtained, its robust and efficient utilisation within process modelling applications may not be straightforward. The solution of conventional (e.g. cubic) EOS posed few problems in this context; however, this is not true for more sophisticated models such as SAFT, for which the evaluation of single-phase properties involves numerical iteration when applied to molecules with association sites.

In the past, process modelling has, to a large extent, been the grateful recipient and user of major advances in physical properties. However, in view of the above considerations, we believe that there are significant benefits to be obtained from the use of process modelling techniques in the development of physical properties technology itself, especially in the areas of model validation, numerical methods and computer implementation and delivery (see, for example, Kakalis et al., 2004). The latter area is already beginning to benefit from recent developments in open software architectures, such as the CAPE-OPEN standard for physical properties reviewed earlier in this paper.

The impact of developments in computer hardware

As process models become more detailed, it is inevitable that model-based calculations will involve increasingly more demanding computations. It is, therefore, highly desirable, and sometimes essential, for process modelling to exploit advances in computer hardware and architecture. Here we focus on two areas, namely computation using multiple processors, and computer memory.

Parallel computation in model-based applications

The emphasis of early work on exploiting computer hardware of advanced architecture for process modelling applications had been in the use of vector supercomputers, the availability of which was rather limited. Much of the research during the past decade, however, has focused on more widely available machines involving multiple processors communicating either via shared memory or message passing based on the widely used PVM and MPI protocols.

As in earlier years, a significant part of the work has been in the area of parallelisation of the linear algebra computations that underpin most types of model-based calculations (see, for example, Mallya et al., 1997, 1999, Camarda and Stadtherr, 1999, Hu et al., 2000). This is understandable in view of the significant part of the computation that is spent in matrix operations.

A second major contributor to the overall computational cost is the evaluation of the residuals of the nonlinear equations in the model and, to a lesser extent, their Jacobian matrices. In principle, this computation can readily be parallelised. Paloschi (1998b) exploited this fact in the context of steady-state simulations in the SPEEDUP® process modelling tool using a quasi-Newton algorithm; the linear algebra computations, making use of an iterative linear solver with an appropriate pre-conditioner, were also parallelised. Paloschi and Zitney (1999) used parallelised residual and Jacobian evaluations for dynamic simulation in SPEEDUP. Similar lines of investigation have also been pursued by Borchardt et al. (1999). More recently, Borchardt (2001) considered system partitioning based on flowsheet connectivity (rather than equations), coupled with the use of block-structured Newton-type methods for the purposes of dynamic simulation.

A characteristic of all the work mentioned above is that it attempts to improve computational speed while maintaining the convergence characteristics, error control and other behaviour relating to reliability and robustness of numerical solution codes. In the authors' opinion, this is a welcome development from some of the earlier work on parallel algorithms (e.g. the waveform relaxation algorithms for dynamic simulation that had received some attention in the 1980s and early 1990s).

On the other hand, the work on parallelised process modelling to date has mostly been limited to academic and/or prototyping efforts. To our knowledge, these have not so far resulted in a parallelised computation capability being delivered as an integral part of a commercial process modelling tool. This contrasts with developments in related software areas, such as CFD and computational chemistry.

One possible reason for this lack of progress is the increasing speed of single processors which allows even complex simulations to be executed within a few minutes. Another reason is the relative lack of regularity of structure (e.g. in comparison with the numerical problems solved by CFD tools); this makes it difficult for parallelised numerical codes (e.g. for linear algebra) to outperform state-of-the-art sequential codes consistently and significantly on machines with relatively small numbers of computer processors. A more practical obstacle may be the fact that the majority of process modelling tools make use of the Microsoft Windows® operating system, while many distributed processor networks currently operate under the Linux operating system.

Nevertheless, there are significant counter-examples to all of the above statements. Moreover, the use of parallel computation becomes more attractive when one considers model-based applications which are more computationally demanding than basic process simulation (see, for example, the work by Keeping and Pantelides, 1998 on the parallelisation of sensitivity calculations for dynamic optimisation) or when one needs to guarantee the global optimality of the solutions of optimisation problems or the identification of all solutions of systems of nonlinear algebraic equations (see the review of parallel branch-and-bound algorithms by Gau and Stadtherr, 2001 and the references therein). As inexpensive multiprocessor machines become more widely available over the next few

years, we believe that the use of parallel computation will become fairly routine, at least for some of the more demanding model-based applications.

Computer memory

The size of mathematical systems of practical interest has been continually increasing over the past decade, both because of the significantly increased level of detail being incorporated within the models of individual unit operations, and because of the wider process envelopes within these unit operations are embedded.

Unsurprisingly, computer memory requirements have joined computational speed as major considerations for process modelling tools. The problem is exacerbated by developments aiming at improved usability (e.g. via graphical user interfaces), better diagnostics during the numerical solution[10], and more well-defined software interfaces and communication between software components[11], all of which pose their own significant demands on memory.

Despite the availability of cheap computer memory hardware, the above considerations have increasingly been leading to a bottleneck which arises primarily from the relatively limited amount of memory that can be addressed using 32 bits. The recent developments towards 64-bit computer architectures are particularly important in this context.

Concluding remarks

It is clear from the review presented in this paper that process modelling has made very significant strides over the past decade. Put quite simply, ten years ago, some of the capabilities that are (almost) routine nowadays were either barely imaginable (e.g. in the area of hybrid process/CFD models) or simply collections of rather vague thoughts and propositions (e.g. on open software architectures for process modelling).

Significant challenges remain in all of the specific areas reviewed, and these have been identified in the corresponding sections of the paper. A more general challenge is how best to allow the incorporation of ideas originating from academic research into software for industrial use. The emergence of open software architectures now provides reasonably straightforward routes for academic developments in some areas, such as physical properties and numerical solvers, to be directly used in commercial process modelling tools. The situation is more problematic in areas of research that are related to the fundamentals of process modelling (cf. the work reviewed in the first section of this paper). Arguably, the task of testing academic ideas and, ultimately, transferring them to commercial use has become more difficult in recent years due to the complexity of modern process modelling software and the degree of advanced software engineering that it entails.

The increasing power of process modelling technology has been bringing new perspectives to the development and deployment of model-based solutions throughout the process lifecycle, from the initial process development to the detailed design of individual items of processing equipment and entire plants, and their control systems. To a large extent, this has been a natural evolution of earlier trends in this area. Perhaps a more interesting development in recent years has been the increasing permeability of the boundary between "off-line" and "on-line" applications. This permeability has two distinct but related positive aspects. First, the process models themselves are re-used for both design and operational tasks. Secondly, standard process modelling software tools are employed for tasks on both sides of the boundary.

The key features of this situation are summarised in figure 4. The off-line activities include the development of the initial model, and its validation against data from laboratory and/or pilot plant experiments, or from industrial plant operations[12]. The resulting validated model forms the basis for off-line activities relating to both design and operation (e.g. operator training); it can also cross the boundary to on-line applications. In this context, it is important for this model to be kept up-to-date by taking account both of external modifications to the plant and of intrinsic changes, such as those arising from heat exchanger fouling or catalyst de-activation. This can be achieved by applying combined data reconciliation and parameter (re-)estimation techniques to plant measurements, which also produces reconciled plant data that can be used for yield accounting and similar applications. The up-to-date model can then be used for a variety of on-line applications, such as supporting the plant personnel in making informed decisions about the plant operation, and in determining optimal steady-state set-points or dynamic operating procedures in view of changing feed stocks, product specifications and demands, and economic parameters.

It is important to emphasise that the above does not necessarily imply that the *same* mathematical model is used for all applications. This may be neither desirable nor feasible in view of the special efficiency and robustness requirements posed by real-time and other applications (e.g. model-predictive control and operator training).

[10] This typically requires maintaining more information during (and, sometimes, after) the numerical solution process than is strictly required by the basic numerical algorithm.

[11] For example, via the use of middleware such as CORBA and COM rather than simple procedure (e.g. subroutine) calls as was the case in the past.

[12] See Pantelides (2001) for a discussion of the importance of this model validation step and its relation to model-based model-targeted experimentation, and a review of some of the relevant literature.

Instead, there may be several models of differing fidelity and behaviour. However, all of them are derived automatically via appropriate manipulations (e.g. state-

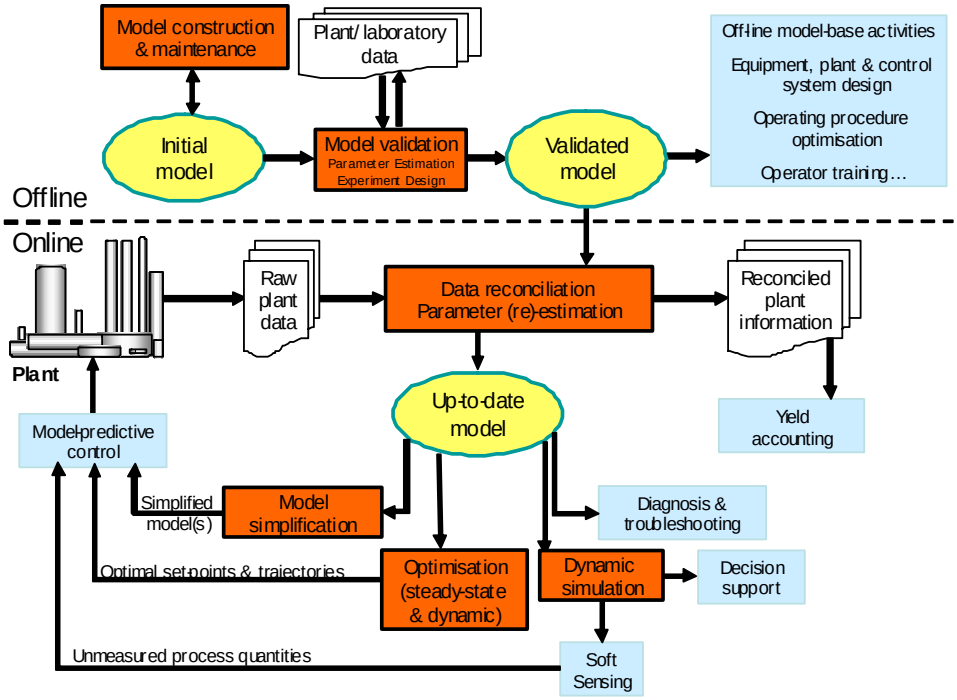

Figure 4. Process models and modelling technology for off-line and on-line applications

space linearisation or nonlinear model reduction) from a single "master" model, thereby ensuring consistency across different applications.

The integrated approach described above can result in significant benefits in terms of the re-usability of process modelling effort and the consistency of the information used throughout the process lifecycle. It also means that any advances in the usability, reliability and efficiency of the underlying process modelling technology become immediately available to all model-based activities and not just to off-line simulations.

Undeniably, the above integrated approach is also particularly challenging, and it has to be acknowledged that the failure to realise even parts of it in the past has led to some understandable scepticism. However, it is also important to recognise that recent progress has brought all of the tasks related to process modelling tools (shown as rectangles with solid borders in figure 4) well within the scope and power of currently available technology. In fact, there are on-going commercial developments in most of the application areas in figure 4 (shown as border-less rectangles), all of them being based on general-purpose process modelling technology rather than on a multitude of specialised software systems as was the case in the past.

Finally, despite this paper's focus on process modelling software, it is worth noting that industrial modelling projects are increasingly interdisciplinary efforts. For example, modelling of fuel cell power plants involves electrical, chemical and control engineers; many projects in the upstream oil industry employ combinations of chemical, civil and mechanical engineers; and modelling projects in the fine chemicals and consumer goods industries require the close collaboration of physicists, chemists and engineers. These trends point towards a unification of mathematical modelling tools across different disciplines of engineering and science, as a means both of facilitating collaboration among these diverse groups, and of minimising the costs associated with software acquisition and deployment. This development is also beneficial to modelling software development, given its ever increasing complexity and cost.

We believe that the advances reviewed in this paper can play a key role towards achieving this unification goal. Of particular importance in this context is the support for formal detailed descriptions of deep domain-specific knowledge of various kinds, coupled with the separation of such content from the modelling software itself. In principle, this allows the same modelling technology to be

used in different domains by providing appropriate domain-specific content.

Acknowledgements

The first author wishes to acknowledge the financial support of the United Kingdom Engineering and Physical Sciences Research Council (EPSRC) to his research on process modelling carried out at Imperial College London under Platform Grant GR/N08636 and Grant GR/N35991. We are also grateful to Duane Griffin, Benjamin Keeping and James Marriott of Process Systems Enterprise Ltd. for many useful discussions on process modelling tools and their architecture.

References

Bauer, M., and G. Eigenberger (1999). A concept for multi-scale modeling of bubble columns and loop reactors. *Chem. Eng. Sci*., **54**, 5109-5117.

Bauer, M., and G. Eigenberger (2001). Multiscale modeling of hydrodynamics, mass transfer and reaction in bubble column reactors. *Chem. Eng. Sci*., **56**, 1067-1074.

Bezzo, F. (2002). *Design of a General Architecture for the Integration of Process Engineering Simulation and Computational Fluid Dynamics*. PhD thesis, University of London, United Kingdom.

Bezzo, F., and S. Macchietto (2004). A general methodology for hybrid multizonal/CFD models. Part II. Automatic zoning. *Comput. chem. Engng*., **28**, 513-525.

Bezzo, F., S. Macchietto, and C.C. Pantelides (2000). A general framework for the integration of computational fluid dynamics and process simulation. *Comput. chem. Engng*., **24**, 653-658.

Bezzo, F., S. Macchietto, and C.C. Pantelides (2003). General hybrid multizonal/CFD approach for bioreactor modeling. *AIChE J*., **49**, 2133-2148.

Bezzo, F., S. Macchietto, and C.C. Pantelides (2004). A general methodology for hybrid multizonal/CFD models. Part I. Theoretical framework, *Comput. chem. Engng*., **28**, 501-511.

Bogusch, R., B. Lohmann, and W. Marquardt (2001). Computer-aided process modeling with MODKIT. *Comput. chem. Engng*., **25**, 963-995.

Bokis, C.P., S. Ramanathan, J. Franjione, A. Buchelli, M.L. Call, and A.L. Brown (2002). Physical properties, reactor modeling and polymerization kinetics in the low-density polyethylene tubular reactor process. *Ind. Eng. Chem. Res.*, **41**, 1017-1030.

Borchardt, J. (2001). Newton-type decomposition methods in large-scale dynamic process simulation. *Comput. chem. Engng*., **25**, 951-961.

Borchardt, J., K. Ehrhardt, F. Grund, and D. Horn (1999). Parallel modular dynamic process simulation. In F. Keil, W. Mackens, H. Voss and J. Werther (Eds.), *Scientific Computing in Chemical Engineering II*, Springer, 152-159.

Braunschweig, B.L., C.C. Pantelides, H. Britt, and S. Sama, (2000). Process modeling: the promise of open software architectures. *Chem. Eng. Progr*., **96**(9), 65-76.

Camarda, K.V., and M.A. Stadtherr (1999). Matrix ordering strategies for process engineering: graph partitioning algorithms for parallel computation. *Comput. chem. Engng*., **23**, 1063-1073.

Chapman, W. G., K.E. Gubbins, G. Jackson, and M. Radosz (1989). SAFT: equation-of-state solution model for associating fluids. *Fluid Phase Equilibria*, **52**, 31-38.

Cheluget, E.L., C.P. Bokis, L. Wardhaugh, C.C. Chen, and J. Fisher (2002). Modeling polyethylene fractionation using the perturbed-chain statistical associating fluid theory equation of state. *Ind. Eng. Chem. Res.*, **41**, 968-988.

Dieterich, E.E., A. Salden, J. Schäfer, J. Schmidt, and G. Eigenberger (1997). Computer aided modeling with BITMAP. *Comput. chem. Engng*., **21**, 1191-1201.

Drengstig, T., S.O. Wasbø, and B. Foss (1997). A formal graphical-based process modeling methodology. *Comput. chem. Engng*., **21S**, S835-S840.

Economou, I. G. (2002). Statistical Associating Fluid Theory: a successful model for the calculation of thermodynamic and phase equilibrium properties of complex fluid mixtures. *Ind. Eng. Chem. Res.*, **41**, 953-962.

Falcitelli M., S. Pasini and L. Tognotti (2002). Modelling practical combustion systems and predicting NOx emissions with an integrated CFD based approach. *Comput. chem. Engng*. **26**, 1171-1183.

Fletcher R., and W. Morton (2000). Initialising distillation column models. *Comput. chem. Engng*. **23**, 1811-1824.

Galindo, A., S.J. Burton, G. Jackson, D.P. Visco, and D.A. Kofke (2002). Improved models for the phase behaviour of hydrogen fluoride: chain and ring aggregates in the SAFT approach and the AEOS model. *Molecular Physics*, **100**, 2241-2259.

Galindo, A., P.J. Whitehead, G. Jackson, and A.N. Burgess (1997). Predicting the phase equilibria of mixtures of hydrogen fluoride with water, difluoromethane (HFC-32), and 1,1,1,2-tetrafluoroethane (HFC-134a) using a simplified SAFT approach. *J. Phys. Chem. B*, **101**, 2082-2091.

Gani, R., and J.P. O'Connell (2001). Properties and CAPE: from present uses to future challenges. *Comput. chem. Engng*., **25**, 3-14.

Gani, R., and E.N. Pistikopoulos (2002). Property modelling and simulation for product and process design. *Fluid Phase Equilibria*, **194-197**, 43-59.

Gau, C.Y., and M.A. Stadtherr (2001). Parallel branch-and-bound for chemical engineering applications: load balancing and scheduling issues. *Lecture Notes in Computer Science*, **1981**, 273-300.

Gau C.Y., and M.A. Stadtherr (2002). New interval methodologies for reliable chemical process modeling. *Comput. chem. Engng*., **26**, 827-840.

Gil-Villegas, A., A. Galindo, P.J. Whitehead, S.J. Mills, G. Jackson, and A.N. Burgess (1997). Statistical Associating Fluid Theory for potentials of variable range. *J. Chem. Phys.*, **106**, 4168-4186.

Gil-Villegas, A., A. Galindo, and G. Jackson (2001). A statistical associating fluid theory for electrolyte solutions (SAFT-VRE). *Molecular Physics*, **99**, 531-536.

Gross, J., and G. Sadowski (2001). Perturbed-chain SAFT: an equation of state based on perturbation theory for chain molecules. *Ind. Eng. Chem. Res.*, **40**, 1244-1260.

Gross, J., and G. Sadowski (2002). Application of the perturbed-chain SAFT equation of state to associating systems. *Ind. Eng. Chem. Res.*, **41**, 5510-5518.

Hu, Y-F., K.C.F. Maguire, and R.J. Blake (2000). A multilevel unsymmetric matrix ordering algorithm for parallel process simulation. *Comput. chem. Engng*. **23**, 1631-1647.

Huang, S. H., and M. Radosz (1990). Equation of state for small, large, polydispersive and associating molecules. *Ind. Eng. Chem. Res.*, **29**, 2284-2294.

Huang, S. H., and M. Radosz (1991). Equation of state for small, large, polydispersive and associating molecules: extension to fluid mixtures. *Ind. Eng. Chem. Res.*, **30**, 1994-2005.

Jensen, A.K., and R. Gani (1996). A computer-aided system for generation of problem specific process models. *Comput. chem. Engng*., **20S**, S145-S150.

Kakalis, N.M.P., A.I. Kakhu, and C.C. Pantelides (2004). Implementing SAFT-based thermodynamics in process modelling tools. In C.A. Floudas and R. Agrawal (Eds.), *Proc. Conf. on Foundations of Computer-Aided Process Design*, CACHE Publications.

Kakhu, A.I., B.R. Keeping, Y. Lu, and C.C. Pantelides (1998). An open software architecture for process modelling and model-based applications. In J. Pekny and G. Blau (Eds.), *Proc. 3rd Conf. on Foundations of Computer-Aided Process Operations*, CACHE Publications, 518-524.

Keeping, B.R, and C.C. Pantelides (1998). A distributed memory parallel algorithm for the efficient computation of sensitivities of differential-algebraic systems. *Mathematics and Computers in Simulation*, **44**, 545-558.

Linninger, A.A., S. Chowdhry, V. Bahl, H. Krendl and H. Pinger (2000). A systems approach for mathematical modeling of industrial processes. *Comput. chem. Engng*., **24**, 591-598.

Linninger, A.A., and H. Krendl (1999). TechTool - Computer-aided generation of process models (Part 1 - A generic mathematical language). *Comput. chem. Engng*., **23S**, S703-S706.

Mallya, J.U., S.E. Zitney, S. Choudhary, and M.A. Stadtherr (1997). A parallel frontal solver for large scale process simulation and optimization. *AIChE J*., **43**, 1032-1040.

Mallya, J.U., S.E. Zitney, S. Choudhary, and M.A. Stadtherr (1999). Matrix reordering effects on a parallel frontal solver for large scale process simulation. *Comput. chem. Engng*., **23**, 585-593.

Marias, F. (2003). A model of a rotary kiln incinerator including processes occurring within the solid and the gaseous phases. *Comput. chem. Engng*., **27**, 813-825.

Martinson, W.S. (2000). *Index and Characteristic Analysis of Partial Differential Equations.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Martinson, W.S., and P.I. Barton (2001). Index and characteristic analysis of linear PDAE systems. *SIAM. J. Sci. Comput*., **24**, 905-923.

Müller, E. A., and K.E. Gubbins (2001). Molecular-based equations of state for associating fluids: a review of SAFT and related approaches. *Ind. Eng. Chem. Res.*, **40**, 2193-2211.

Neumann, J. (2004). *Consistency Analysis of Systems of Partial and Ordinary Differential and Algebraic Equations*, PhD thesis, University of London, United Kingdom.

Oh, M., and C.C. Pantelides (1996). A modelling and simulation language for combined lumped and distributed parameter systems. *Comput. chem. Engng*., **20**, 611-633.

Paloschi, J.R. (1995). Bounded homotopies to solve systems of algebraic nonlinear equations. *Comput. chem. Engng*., **19**, 1243-1254.

Paloschi, J.R. (1997). Bounded homotopies to solve systems of sparse algebraic nonlinear equations. *Comput. chem. Engng*., **21**, 531-541.

Paloschi, J.R. (1998a). Using sparse bounded homotopies in the SPEEDUP simulation package. *Comput. chem. Engng*., **22**, 1181-1188.

Paloschi, J.R. (1998b). Steps towards steady-state process simulation on MIMD machines: implementation in the SPEEDUP simulator. *Comput. chem. Engng*., **22**, 1189-1196.

Paloschi, J.R., and S. E. Zitney (1999). Parallel dynamic simulation of industrial chemical processes on distributed memory computers. *Comput. chem. Engng*., **23S**, S395-S398.

Pantelides, C.C. (2001). New challenges and opportunities for process modelling. In R. Gani, and S. Bay-Jørgensen (Eds.), *Proc. of ESCAPE'11 Conference*, Elsevier, 15-26.

Pantelides, C.C., and H.I. Britt (1995). Multipurpose process modeling environments. In L.T. Biegler and M.F. Doherty (Eds.), *Proc. Conf. on Foundations of Computer-Aided Process Design'94*, CACHE Publications , 128-141.

Paricaud P., A. Galindo, and G. Jackson (2002). Recent advances in the use of the SAFT approach in describing electrolytes, interfaces, liquid crystals and polymers. *Fluid Phase Equilibria*, **194**, 87-96.

Patel B.H., P. Paricaud, A. Galindo, and G.C. Maitland (2003). Prediction of the salting-out effect of strong electrolytes on water plus alkane solutions. *Ind. Eng. Chem. Res*., **42**, 3809-3823.

Perkins, J.D., R.W.H. Sargent, R. Vázquez-Román, and J.H. Cho (1996). Computer generation of process models. *Comput. chem. Engng*., **20**, 635-640.

Process Systems Enterprise Ltd. (2003). *gO:CFD: gPROMS Object for Reactive CFD*. London, United Kingdom.

Sadowski, G. (2004). Thermodynamics of polymer systems. *Macromolecular Symposia*, **206**, 333-346.

Schnepper, C.A,. and M.A. Stadtherr (1996). Robust process simulation using interval methods. *Comput. chem. Engng*., **20**, 187-199.

Urban, Z., and L. Liberis (1999). Hybrid gPROMS-CFD modelling of an industrial-scale crystalliser with rigorous crystal nucleation and growth kinetics and a full population balance. In *Proc. Chemputers 1999*, Düsseldorf, Germany.

Urban, Z., R. Cettour, and C.C. Pantelides (2003). High-accuracy predictive modelling for solid oxide fuel cell units and systems. In *Proc. Eighth Grove Fuel Cell Symposium*, 24-26 September 2003, London, United Kingdom.

Westerweele, M.R. (2003). *Five Steps for Building Consistent Dynamic Process Models and their Implementation in the Computer Tool Modeller*. PhD thesis, University of Eindhoven, the Netherlands.

Westerweele, M.R., H.A. Preisig and M. Weiss (1999). A systems approach for mathematical modeling of industrial processes. *Comput. chem. Engng*., **23S**, S751-S754.