Towards computer-aided multiscale modelling: an overarching methodology and support of conceptual modelling

Yang Zhao, Cheng Jiang, Aidong Yang*

Division of Civil, Chemical and Environmental Engineering, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford GU2 7XH, UK

Abstract

Multiscale modelling is now widely regarded as a promising and powerful tool in various disciplines, including the broad area of process engineering. However, a multiscale model is usually much more difficult to develop than a single-scale model due to a range of conceptual, numerical, and software challenges. Currently, there is little support developed to facilitate multiscale modelling. This paper discusses the key challenges faced by computer-aided multiscale modelling (CAMM) and presents a methodology for developing a computer-based, generic and open supporting framework for multiscale modelling. Details are particularly provided on the development of a conceptual modelling tool, an important element of the envisaged tool set for CAMM. The application of this tool is illustrated by two reactor modelling examples.

Keywords: multiscale modelling, computer aided modelling, conceptual modelling

* Corresponding author. Tel.:+44 1483 686577; fax: +44 1483 6581. E-mail address: a.yang@surrey.ac.uk (A. Yang).

1. Introduction

Multiscale modelling is an emerging modelling paradigm which combines the models of different resolution scales of a complex system to obtain a characterisation of the system for a better model quality or for a higher computational efficiency. This modelling paradigm is now widely regarded as a promising and powerful tool in various disciplines. In process engineering, typical applications in the design of chemical processes and products have been presented in several reviews (e.g. Braatz et al., 2004; Charpentier, 2002, 2009; Lucia, 2010; Vlachos, 2005). In material science, this modelling paradigm has been applied for predicting properties, functionalities, and the formation and propagation of defect, with the combination of quantum mechanics, molecular dynamics and finite element modelling (Karaksidis and Charitidis, 2007; Marouds, 2000). In computational mechanics and particularly fluids modelling, a representative application is to combine molecular dynamics with continuum modelling to capture the behaviour at the near-wall region (Liu et al., 2007; O'Connell and Thompson, 1995). In systems biology, this multiscale paradigm has been applied to the modelling of ion channels (quantum-molecule-cell) and heart (cell-tissue-organ) (Southern et al., 2008).

Developing a multiscale model is generally much more challenging than building a single scale model, because the modeller has to determine what scales to be involved and how the involved scales should be connected. To tackle the difficulties, several efforts in formulating modelling methodologies have aimed to provide generic guidance to the development of new applications. The most notable activities are devoted to the classification of multiscale modelling approaches, frameworks, or model types which define the generic ways how submodels of different scales of a system can be introduced and coupled to form an integral model. Pantelides (2001) classifies scale integration strategies into serial, parallel, hierarchical, and simultaneous approaches. This classification was expanded and adapted in the work of Ingram et al. (2004) which identifies a number of integration frameworks and discusses the compatibility issue. Li et al. (2005) articulate the difference between what is called descriptive, correlative, and variational methods. Vlachos (2005) makes the distinction between sequential or serial approaches and those referred to as hybrid, parallel, dynamic, and concurrent approaches.

While the guidance offered by the generalised modelling methodologies may help the modellers conceptually, computer-based tools aim to provide more direct and concrete support to facilitate modellers in their specific modelling tasks (Ingram et al., 2004; Marquardt et al., 2000). In this direction, the work reported in Bezzo et al. (2004) treats the coupling of a multizonal model of process equipment and a CFD model, which represents a particular type of multiscale modelling tasks in process engineering. Kulikov et al. (2005) present a generic implementation of the coupling of CFD and population balance modelling tools based on an environment for simulation tools integration. Morales-Rodrígeuz and Gani (2007) report the integration of several, scale-crossing modelling tools to support product-centric process design and analysis. Apart from the efforts made within the process systems engineering community, Doi (2002) reports a material modelling platform to support multiscale modelling of polymers by integrating a specific set of simulation programmes each of which is dedicated to the modelling at a certain length scale. In structural biology, the MMTSB tool set (Feig et al., 2004) has been developed which is capable of supporting tasks such as translation between all-atom and low-resolution models. While these efforts have all provided useful support to specific types of multiscale modelling applications, they are commonly intended to address specific types of applications.

Recognising that multiscale modelling is a generic modelling paradigm applicable to various disciplines, this current work has attempted to explore a way of providing a generic set of tools to support multiscale modelling in different applications. The ultimate goal is to build a generic and open environment for computer-aided multiscale modelling (CAMM), which on the one hand provides common facilities sharable by different modelling tasks (possibly from different disciplines) and on the other hand offers a mechanism for the modellers to incorporate task- or domain-specific knowledge and information and marry it with the common facilities. In the remainder of this paper, Section 2 identifies the key challenges for CAMM as well as the basis on which the proposed research on CAMM could build. Section 3 outlines an overarching methodology for developing a supporting environment for CAMM as envisaged above. In Section 4, the development and application examples of a conceptual modelling tool, an important element in the generic tool set following the methodology proposed, are presented. Some concluding remarks, including those on the development of the other elements of the proposed tool set, are given in Section 5.

2. Computer-aided multiscale modelling: Starting point and key challenges

From the perspective of process systems engineering, CAMM as a relatively new research area is naturally to be based on computer-aided process modelling (CAPM), which has gained significant development over the past several decades. This includes a number of commercially highly successful software packages for process modelling, simulation, and optimization. In academic research, an important methodological development in CAPM has been phenomena-based modelling, which is aimed at reducing the effort of human modellers by allowing them to work with process engineering concepts instead of mathematical details, as advocated by e.g. Stephanopoulos et al. (1990), Marquardt (1995), Perkins et al. (1996), Jensen and Gani (1996), Westerweele et al. (1999), Bieszczad (2000), and Bogusch et al. (2001). More recently, ontologies, i.e. explicit specifications of conceptualisation of domains of interest in terms of definitions of concepts and their relations (Gruber, 1993), have been applied in the field of process modelling. Batres et al. (2002) apply ontologies to describe meta-models which are the representations of simulation models of physicochemical behaviour. Yang et al. (2004) propose an ontology-based modelling approach to improve the generality and extensibility of phenomena-based modelling tools. A four-level architecture for a collaborative environment to support knowledge sharing in industrial ecology has been introduced by Kraines et al. (2005) in which ontologies play the role of formally representing computational models.

When the simulation of a complex process system has to be accomplished using multiple simulation tools, a component-based approach has been adopted by the process engineering community: the CAPE-OPEN initiative (Braunschweig et al., 2000) develops standards of software interfaces for typical process modelling components. The result of CAPE-OPEN has been vital for the improvement of openness in commercial process modelling software, and for the development of open simulation platforms such as CHEOPS (Schopfer et al., 2004). In some other areas such as biological systems modelling (e.g. Hetherington et al., 2007; Hunter et al., 2005) and multi-physics modelling (e.g. Smirnov, 2004), there have been similar developments in supporting the combined use of different models and simulation tools.

Starting with the relevant development of CAPM as outlined above, two different sets of challenging issues must be addressed in the process of marching towards CAMM. Within the *first* set, there are conceptual, numerical, and software implementation related issues which will be faced by modellers in a specific multiscale modelling application:

- *Conceptually*, and usually at the early stage of developing a multiscale model for a specific system/problem, the modeller has to decide what scales of the system or the subsystems should be considered, what characteristics of each of these scales should be modelled, and what connections should be established between these scales and between subsystems. Essentially "designing" the multiscale model, it often proves to be a nontrivial task especially for an inexperienced modeller.
- *Numerically*, one may expect that coupled simulation codes for different scales will run stably if each of these codes is numerically stable when being executed individually. This is however not necessarily the case (Bezzo et al., 2005; Braatz et al., 2006; Pantelides, 2001, Rusli et al., 2004). Numerical instability may arise in multiscale modelling due to e.g. temporal/spatial mismatch between the solutions at different scales or noises in the data passed between scales. Furthermore, modelling errors may occur as the consequence of handling numerical instability by means of filtering, or due to the aggregation and disaggregation of the information between different scales. To render successful

coupling, one should be aware of the numerical consequences of implementing a particular coupling scheme, and apply appropriate methods to suppress numerical instability and to reduce modelling errors whenever possible.

- *With respect to software implementation*, a common issue in multiscale modelling is that different specialty modelling tools may be used for different scales of a system; these tools will have to be integrated to allow a multiscale model to be solved or executed as a whole. As these tools have usually been developed separately by different parties, significant heterogeneity often exists in their data structures, interfaces, and even supported operating systems, making the integration of these tools a very challenging task.

Additionally, there is a *second* set of challenging issues which are concerned with the generality and "powerfulness" of the computer-based support to multiscale modelling. Briefly, this set of challenges are about providing answers to the following two questions:

- *How to maximise the support to modellers at each step of model development*, given the very different nature of the conceptual, numerical, and software related problems to be tackled and given the varying levels of expertise possessed by different modellers?
- *How to make a computer-aided modelling system as generic as possible*, in light of the highly diverse nature of various multiscale modelling applications?

3. An overarching methodology for CAMM

To effectively address the aforementioned challenges, an overarching methodology with two different aspects is proposed (cf. Fig. 1). These two aspects are presented separately in the subsections below. Given the complexity of the issues faced by CAMM, this methodology is not intended to provide solutions to all the challenges. However, it does directly address the second set of issues mentioned above. Besides, the methodology is also intended to offer an overarching framework in which solutions to the first set of issues can be accommodated.

3.1 Establishing a hierarchical theoretical framework of multiscale systems

A unified theoretical framework may be established to cope with the diversity of applications that a generic tool set for CAMM should support. More specifically, this framework may possess a hierarchical structure comprising a *fundamental level* and a *domain-specific level*. The former aims to provide *a unified theory for general multiscale systems*. As demonstrated in Yang and Marquardt (2009), this may be achieved by a *deductive* approach on the basis of general systems theory and its formal formulations, allowing the formulation of explicit and rigorous definitions of fundamental concepts in multiscale modelling. These concepts may include (i) system, subsystem, environment, inter-subsystem relation, system-environment relation; (ii) scale, inter-scale relation*,* and (iii) property and law (set) of all the above items. From these concepts and particularly the inter-subsystem and inter-scale relations, a variety of generic multiscale modelling schemes or model types may be derived naturally and rigorously.

On top of this *fundamental level*, multiscale concepts for specific domains can be introduced. For the domain of process engineering, this level aims to provide generic definitions of scales typically involved in the multiscale modelling of process systems, such as molecules, molecule clusters, particles, surfaces and phases, process units, plants, sites, etc. An existing conceptualization of chemical process systems such as OntoCAPE (Marquardt et al., 2010; Morbach et al., 2007) may be reused for this purpose.

It is expected that, by building upon this unified, hierarchical theoretical framework, the generality of the CAMM solutions can be maximised: When a method for a particular problem (be it a conceptual, numerical, or software-related one) is developed, one can always try to base it, in the first place, only on the theory of general multiscale systems (i.e. the fundamental level). If it is successful, the resulting method will be applicable to all kinds of applications as long as the subjects of modelling conform to this theory. If the method has to exploit domain-specific (process engineering for instance) concepts, the generic theory for multiscale process systems at the *domain-specific level* will come to play. This will ensure that this method be applicable to various process modelling applications.

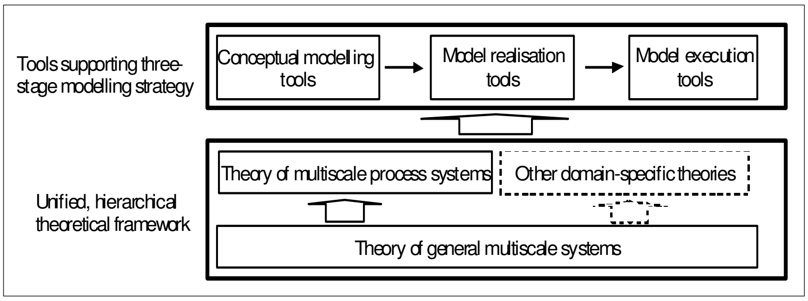

Tools supporting three-stage modelling strategy
Conceptual modelling tools
Model realisation tools
Model execution tools
Unified, hierarchical theoretical framework
Theory of multiscale process systems
Other domain-specific theories
Theory of general multiscale systems

Figure 1. An overarching methodology for computer-aided multiscale modelling.

3.2 Applying a three-stage strategy

Motivated by the need of addressing different types of modelling issues while maximising computer-based support to human modellers, it would be instrumental to separate the most creative modelling tasks which inevitably require the input/intervention from the human modellers (referred to as Group 1), from other tasks which can be handled more "automatically" by software tools (referred to as Group 2). For this purpose, activities in a multiscale modelling application may be organized into three successive stages: *conceptual modelling*, *model realisation*, and *model execution*. For a given modelling problem, the first stage results in a *conceptual model* which dictates (by "words") what scales to be included in the model and what kind of linkages should be established between these scales. This is a stage which mainly addresses the aforementioned *conceptual* challenges. Largely belonging to Group 1, the tasks in conceptual modelling can hardly be fully *automated*; they will rather be *supported* through the interactions between a computer-based conceptual modelling tool and its users. Following this stage, the model realisation and execution stages deal with *(i)* the realisation of the conceptual model (to a form which is ready to execute) and *(ii)* the actual execution of model, respectively. The modelling tasks to be tackled in these two stages largely belong to Group 2, because they involve predominantly numerical and software implementation related work as opposed to conceptual work. This type of work may be completed mainly by using designated software tools that implement numerical procedures for handling issues commonly encountered in multiscale modelling, hence requiring minimum effort from the human modeller.

3.3 Comments

As aforementioned, the methodology for CAMM proposed above aims to maximises the generality of the CAMM tools and to enhance computer-based support. Furthermore, it provides an overarching framework in which solutions for specific conceptual, numerical and software implementation issues can be developed. However, this methodology does not replace methodologies for supporting specific modelling stages. In Section 4, the methodology for supporting the first modelling stage, namely conceptual modelling and its implementation will be presented.

4. Supporting conceptual modelling

As discussed earlier, a conceptual model describes what a (mathematical) multiscale model should contain in terms of the scales involved, the composition of each scale, and the connections between different scales. A conceptual model is represented using the concepts defined in the unified, hierarchical framework of multiscale modelling theories, and may be utilized in two ways. When a multiscale model is to be developed from scratch and can be constructed (in an ideal case) in the form of a single set of equations, automatic model generation may be carried out. This process essentially assembles, according to the content of the conceptual model, a composite model with components from a library of elementary models. The principle of this type of model generation has been explored by previous research in phenomena-based modelling (e.g. Bieszczad, 2000; Yang et al., 2004). In the cases where a multiscale model is to be realised through the integration of existing models implemented in different tools, a conceptual model may be used to guide the integration and to facilitate the execution of a "heterogeneous" multiscale simulation; an initial exploration in this direction is reported in Zhao et al. (2011). Besides, one can envisage a common usage of a conceptual model, which is to provide a well-structured documentation of the corresponding multiscale model, supporting future extension, adaptation, and reuse.

Conceptual modelling is a task difficult to be done automatically by computer alone. However, computer-based facilitation may be provided to support users in the course of constructing conceptual models. In this section, the methodology for developing a computer-aided conceptual modelling tool (CCMT) and the implementation of the methodology are first presented. The application of the CCMT is subsequently illustrated by two examples.

4.1. Methodology for developing the CCMT

The CCMT has been developed by adopting a hierarchical conceptualisation of multiscale systems, based on which the tool is designed to offer desired functions. Details of these aspects are presented in the two subsections below.

4.1.1. Ontology based representation of a hierarchical conceptualization

As discussed in Section 3.2, the generality of tools for CAMM (including the CCMT) is to be achieved by adopting a generic conceptual foundation, which forms the basis of developing the conceptualization for multiscale systems in various domains. To this end, a generic ontology, explicitly specifying such a generic conceptual foundation, is developed based on the work of Yang and Marquardt (2009). The conceptualisation captured by this ontology was originally developed on the basis of general systems theory. In the past, several other approaches have been adopted for formulating abstract (and common) representations of physical systems, e.g. the network model based approach (Mangold et al., 2002) and the bond graph model based approach (Couenne et al., 2006; Couenne et al., 2008). In the present work, the conceptualisation by Yang and Marquardt (2009) has been adopted due to its generality as well as its provision of the concepts required for multiscale modelling.

Figure 2 shows the important elements of the generic ontology adopted in this work in the form of a UML class diagram. Starting with very fundamental concepts, the ontology states that a *thing* has *states*, each characterised by a number of *state functions* that may have values dependent of the backdrop (temporal and spatial locations, etc.). A *thing* may have a number of *phenomena*, each concerning some *state functions* which are governed by certain *laws*. The concept of a *system* is then introduced through defining *couplings* between individual *things*. The existence of *couplings* around a *thing* leads to the classification of its *state functions* into two categories, namely *coupling-induced state functions* (i.e. those that occur due to the existence of a coupling, e.g. properties of a flux that goes across the boundary of a thing) and *non coupling induced state functions*. A *coupling* may be characterised by a *coupling mechanism law* (e.g. the Fick's law for mass diffusion between two parts of a system) and a *topological connection law* (e.g. one that equates an outlet flux of one part to an inlet flux of another). Furthermore, a *system* may have multiple *scales*, each of which is composed of connected components located at a specific level of decomposition. Two different *scales* may be related through an *interscale link*, which may be parallel (in case of two-way exchange of information) or serial (in case of one-way exchange of information). An *interscale link* may be quantified by an *aggregation law*, a *disaggregation law*, and/or a *mereological connection law*. The first two laws quantify how aggregation and disaggregation of quantities (state functions) are carried out, while the third law defines how a coupling at a higher scale is mapped to one or more couplings at a lower scale. A more rigorous and complete description of the ontology can be found in Yang and Marquardt (2009) and in Zhao (2010).

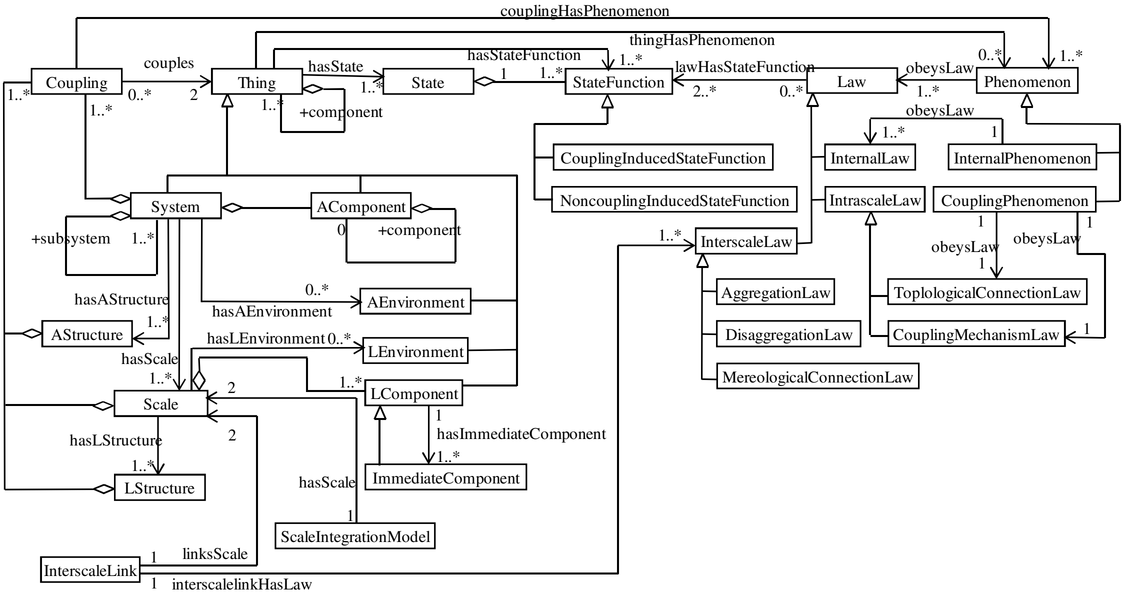


Figure 2. The UML class diagram of the generic ontology adopted in this work.

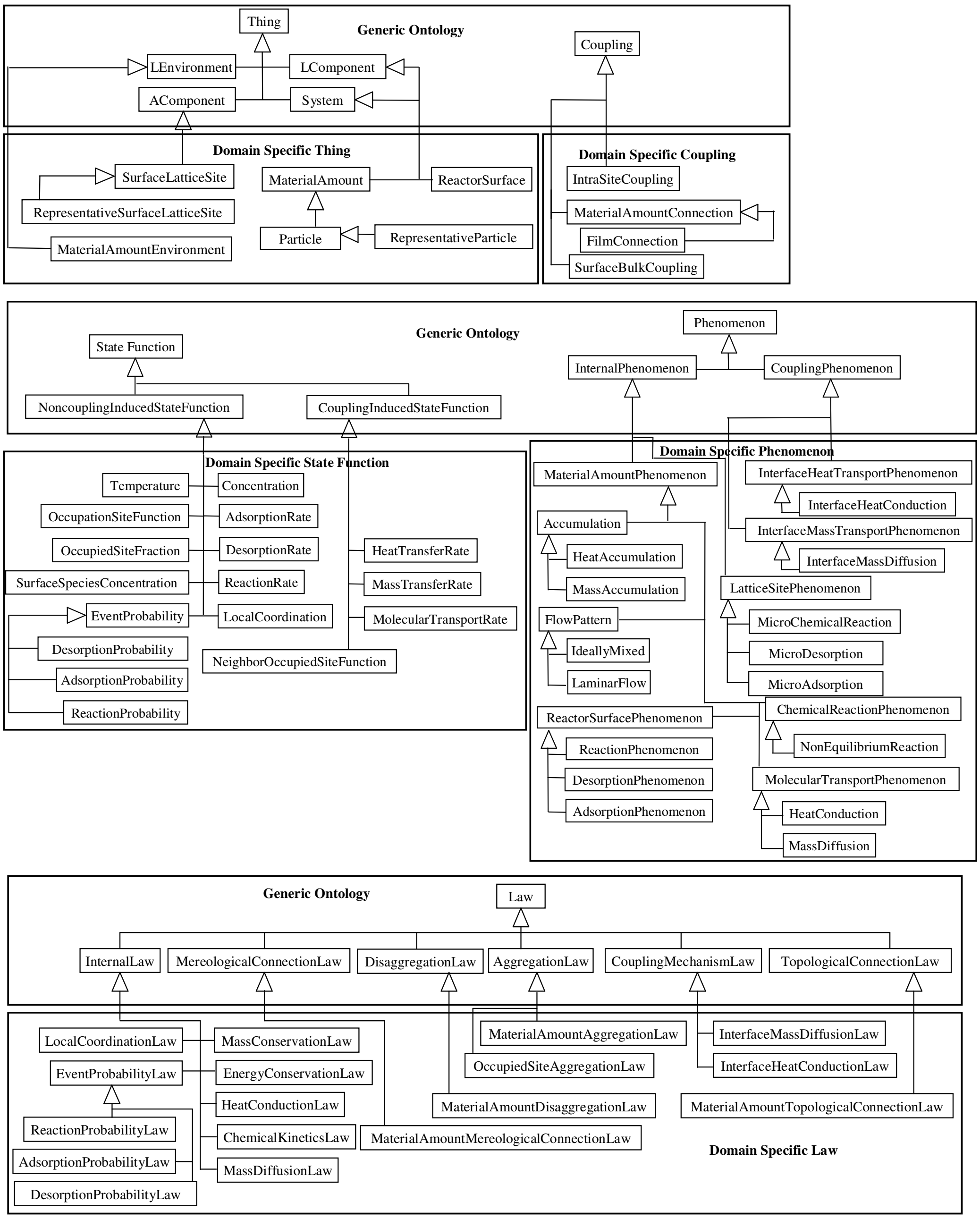


Figure 3. The UML class diagrams of a (partial) process engineering domain ontology and its relation to the generic ontology.

The above-mentioned generic ontology for multiscale modelling constitutes the fundamental level of the unified conceptualisation required as part of the overarching methodology. On top of this, the domain-specific level can be introduced in the form of domain ontologies. Figure 3 illustrates, again in the form of UML class diagrams, how a small part of OntoCAPE is adapted and extended to provide a domain ontology for the application examples which will be presented later in Section 4.3.

In this work, both the generic ontology and domain ontologies are implemented using the formal ontology language OWL (http://www.w3.org/TR/owl-ref/), which can be processed by software tools to support reasoning and other functions to be offered by the CCMT.

4.1.2. Structure and functions of the CCMT

The design of the CCMT is illustrated in Figure 4. Based on the generic ontology of multiscale systems, the tool assumes the conceptual description of any multiscale system will conform at the abstract level to what is defined in the generic ontology. On the other hand, it is expected that it is the domain-specific concepts that should be directly used for constructing a specific conceptual model for an application of a particular domain. For example, for modelling a chemical process, the terms used could be phases, process units, plants, etc. whereas in the case of modelling a biological system concepts such as cells, tissues, and organs may be used instead. Following the principle discussed earlier in Section 3.1, this requirement is met by introducing a domain ontology (representing a domain-specific theory for multiscale modelling) which is a specialisation of the generic ontology (representing a generic theory for multiscale modelling). The central idea here is that the CCMT is developed so as to be able to read a domain ontology and use its content to support a particular conceptual modelling task, as long as the domain ontology is derived on top of the generic ontology.

Within the CCMT, three types of functions are provided through a graphical user interface (GUI) and by leveraging generic tools for processing ontologies.

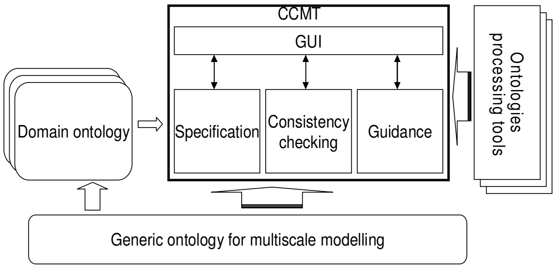

CCMT
GUI
Domain ontology
Specification
Consistency checking
Guidance
Ontologies processing tools
Generic ontology for multiscale modelling

Figure 4. Design of the computer-aided conceptual modelling tool (CCMT)

Specification. This basic function supports the users in specifying the conceptual model in their mind in a structured way, i.e. by describing individual scales and the relationships between any coupled scales. This includes creating/deleting instances of domain specific concepts and specifying each instance in detail. The GUI is used to allow the users to perform these tasks interactively with the tool. A general workflow for specifying a conceptual model for a N-scale system is schematically shown in Figure 5.

Consistency checking. This function checks the consistency between the conceptual model generated by the user against the (axiom) definitions in the generic ontology and the domain ontology. The axioms in the ontology are used to formally define concepts, introducing constraints on what kind of relations they may have with other concepts and how many other concepts can be involved in a relation. Checking the conceptual model against these axioms offers the first "safety guard" of the correctness of the multiscale model to be derived later on (see Section 4.3.1.3 for an example of consistency checking).

Guidance. It is a challenging task to provide generic yet useful guidance to the user when s/he works on a modelling task of a particular problem domain. One attempt has been to offer guidance based on some generic model types, e.g. those proposed by Pantelides (2001) and Ingram et al. (2004) and recently redefined by Yang and Marquardt (2009). Figure 6 shows the types of multiscale model already defined as part of the generic ontology. More specifically, two different kinds of support are considered. Firstly, the user is able to choose for the modelling application at hand one of the model types defined in the generic ontology. The CCMT will then be able to check at any point of the modelling process on whether the current model confines to the prescribed model type. This function is essentially a specialisation of consistency checking mentioned earlier. Alternatively, the user may start without pre-determining the model type, however he or she can classify a resulting conceptual model to see which type this model belongs to. This information may be further used by the modeller to assess the implication of his "conceptual design" of the multiscale model or by the tools that support subsequent modelling stages (i.e. model realisation and model execution).

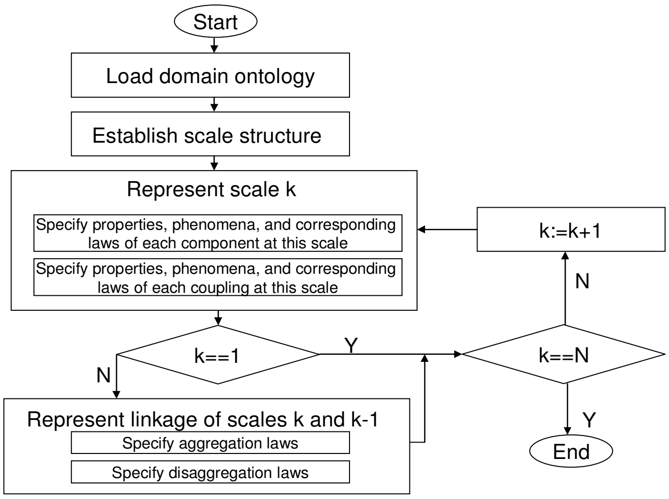

Start
Load domain ontology
Establish scale structure
Represent scale k
Specify properties, phenomena, and corresponding laws of each component at this scale
Specify properties, phenomena, and corresponding laws of each coupling at this scale
k:=k+1
N
k==1
Y
k==N
N
Represent linkage of scales k and k-1
Specify aggregation laws
Specify disaggregation laws
Y
End

Figure 5. A simplified representation of the workflow for conceptual modelling.

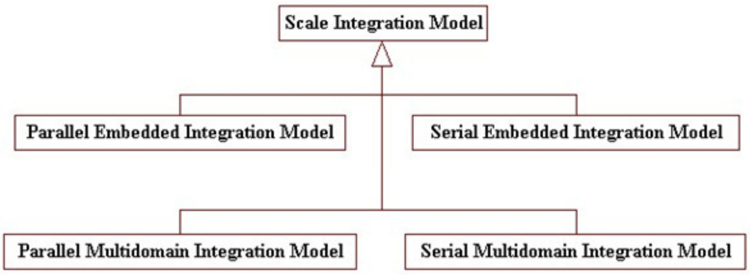
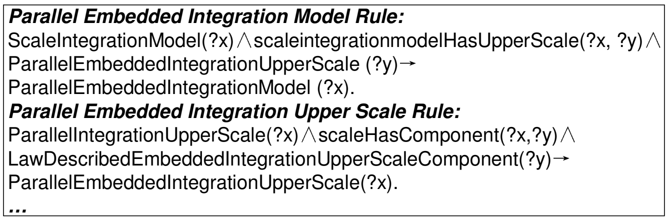

Scale Integration Model
Parallel Embedded Integration Model
Serial Embedded Integration Model
Parallel Multidomain Integration Model
Serial Multidomain Integration Model

Figure 6. Types of multiscale integration models defined in the generic ontology.

4.2. Implementation

A prototype of the CCMT has been implemented using Java and based on Jena (http://jena.sourceforge.net/ontology/), a widely used Java package for processing OWL ontologies. The reasoning capacity required for consistency checking (example in Section 4.3.1.3) and for model type classification (example given in Section 4.3.2.3) is implemented by using the reasoner Pellet (http://clarkparsia.com/pellet/).

The generic ontology has been constructed using Protégé (http://protege.stanford.edu/) with OWL as the ontology modelling language. The generic ontology includes the fundamental concepts as described in Section 4.1.1 as well as the types of multiscale models shown in Figure 6. It is noticed that some of the model types are difficult to define entirely by OWL axioms; in such cases SWRL rules (http://www.w3.org/Submission/SWRL/) are introduced to accomplish the required definitions. Figure 7 illustrates how SWRL rules are used to define a particular modelling type namely “parallel embedded integration model”. Details of a complete description on the ontology classes and SWRL rules introduced for defining the model types are documented in Zhao (2010).



Parallel Embedded Integration Model Rule:
ScaleIntegrationModel(?x)∧scaleintegrationmodelHasUpperScale(?x, ?y)∧
ParallelEmbeddedIntegrationUpperScale (?y)→
ParallelEmbeddedIntegrationModel (?x).
Parallel Embedded Integration Upper Scale Rule:
ParallelIntegrationUpperScale(?x)∧scaleHasComponent(?x,?y)∧
LawDescribedEmbeddedIntegrationUpperScaleComponent(?y)→
ParallelEmbeddedIntegrationUpperScale(?x).
…

Figure 7. Examples of rules introduced for defining multiscale model types.

4.3. Application examples

Two examples are provided here to illustrate the application of the CCMT. The first example is intended to provide details on all the important steps of building a conceptual model by using this tool. With the second example, we will demonstrate how the tool can deal with a different modelling task with a modified domain ontology but without any modification to the tool itself.

4.3.1 Modelling of a packed-bed catalytic reactor

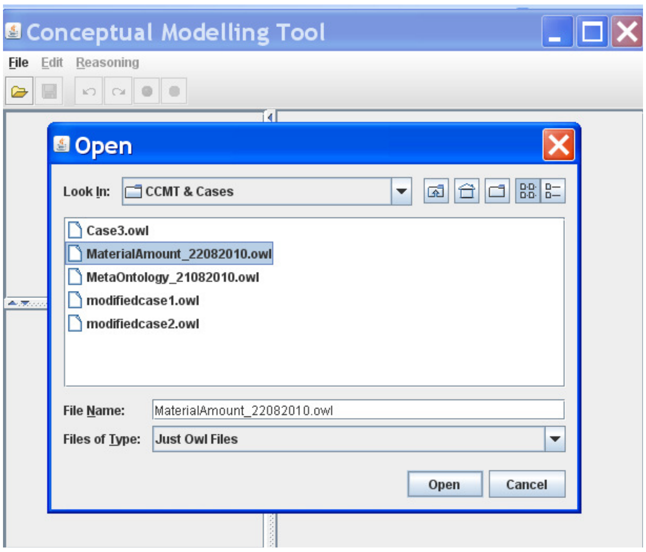
An example of how to build a multiscale conceptual model for a packed-bed catalytic reactor, conforming to a pre-selected model type, will be introduced in this section.

4.3.1.1. The modelling problem

Originally described in Ingram et al. (2004), the packed-bed catalytic reactor is to be modelled at both the bulk phase scale and the catalyst pellet scale, which in this example are referred to as Scale 0 and Scale 1, respectively. Both of the two scales are of a continuum nature but they characterise the reactor at different spatial resolutions. Prior to the construction of a conceptual model for the reactor, it is assumed that a *parallel multi-domain model* is to be established. Formally defined in the generic ontology by means of axioms and rules according to the original definitions by Ingram et al. (2004), the key features of this model type are (i) both aggregation and disaggregation (i.e. two-way) relations exist between two neighbouring scales, and (ii) part of the upper (i.e. coarser) scale does not have internal laws; the modelling of this part of the system relies on the laws introduced at the corresponding part at the lower (i.e. finer) scale. Applying this model type to the example at hand, the Scale 0 of the reactor comprises a bulk fluid phase and a bulk catalyst phase; the latter is modelled through pellets at Scale 1. Furthermore, between the bulk catalyst phase and the pellets exist two-way information flows. These include (i) the aggregation of the temperature and concentration information at the pellet scale to form the corresponding information at the scale of the bulk catalyst phase (required by the modelling of heat and mass transfer at the interface between the two parts at Scale 0), and (ii) the disaggregation (or downward passing) of the (fluid-solid) interface information held by the bulk catalyst phase at Scale 0 to the pellets at Scale 1 which require this information for establishing the boundary conditions.

4.3.1.2 The process of constructing the conceptual model

To use the CCMT to construct such a conceptual model, the modeller starts with loading the domain ontology selected for this task (cf. Figure 8). As aforementioned in Section 4.1.1, this ontology is a small part of OntoCAPE but with proper links with the generic ontology. After that, the modeller represents the intended scale structure by creating two scale instances (“Scale_0” and “Scale_1”) and linking them with an instance of inter-scale link ("InterScaleLink_s0_s1"), as shown in Figure 9. The modeller then has the opportunity to select a model type for this application, where "parallel multi-domain integration model" is chosen as of this example. From this point on, the modeller will be able to invoke the reasoning function of the CCMT to check the compatibility of the conceptual model being constructed to this pre-chosen model type.



File Name: MaterialAmount_22082010.owl

Files of Type: Just Owl Files

Figure 8. Loading a domain ontology into CCMT.

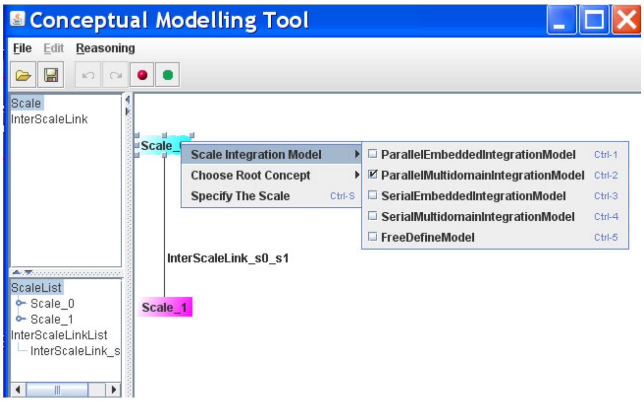


Figure 9. Establishing an interscale link between two scales and choosing a multiscale integration model.

After the overall scale structure is created as described above, the modeller continues to specify each scale involved as well as the link between the two scales. As shown in the "Scale_0" window in Figure

10, the concepts defined in the domain ontology (listed in the left panel of the window) are presented to the modeller; instances of these concepts can be created to describe what is actually included within Scale 0 (shown in the right panel of the window). Here "fluid_phase" and "catalyst_phase" are instances of *material amount*, and "bulk_phase_connection" is an instance of *film connection*. Any of these three elements at this scale can be further specified. For a system component such as the one termed "catalyst_phase", it can be specified in three aspects according to the generic ontology, namely "state function", "internal phenomenon", and "internal law". Figure 11, particularly the left panel of the "catalyst_phase property" window shows the list of state functions loaded from the domain ontology which are applicable to *material amount* (the class of "catalyst_phase"). On the right panel, several instances of these state functions are created by the modeller. In principle, the same kind of specification can be done for *phenomena* and corresponding *laws* of this system component, all according to the relevant concepts loaded from the domain ontology. Figure 12 however shows that for "catalyst_phase", no instances of *phenomena* are to be created because no *laws* (of any phenomena) are to be created for this part of the system at scale 0, according to the decision made earlier. An instance of *coupling* between two system components can be specified in a similar way, i.e. by creating instances of relevant concepts loaded from the domain ontology that belong to several categories as defined in the generic ontology. Figure 13 shows the three categories or classes (*coupling phenomenon*, *coupling mechanism law*, and *topological connection law*, all defined in the generic ontology) and illustrates particularly what instances of *coupling phenomenon* are declared for the coupling between the fluid phase and the catalyst phase (termed "fluid_catalyst_connection").

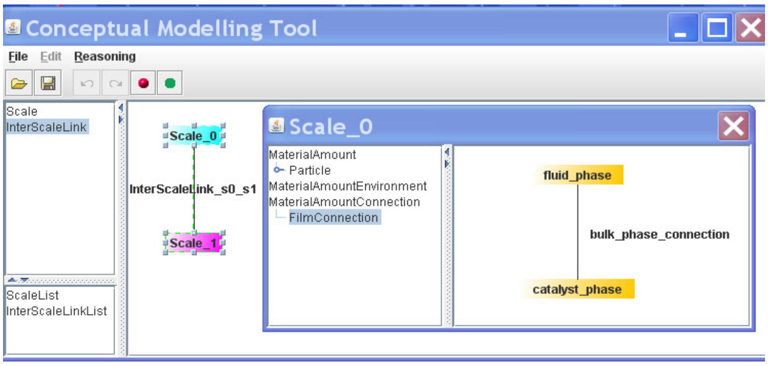


Figure 10. Specification of a scale.

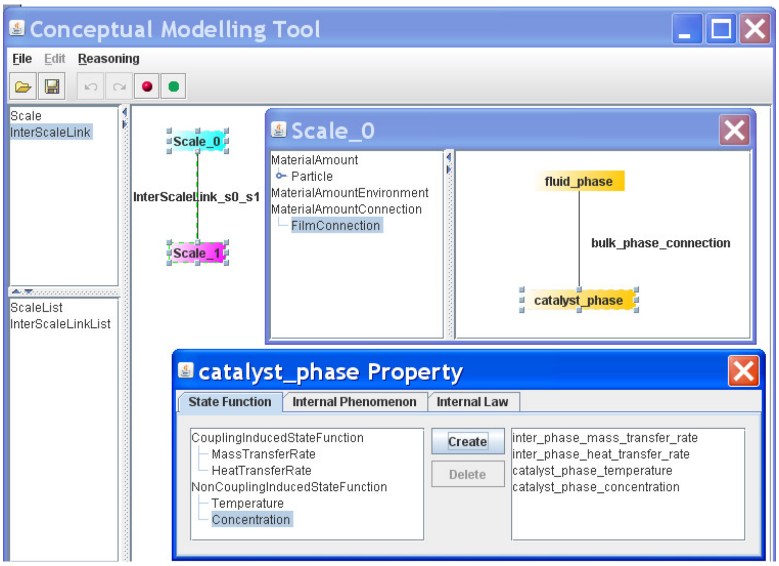


Figure 11. Specifying state functions of a scale component.

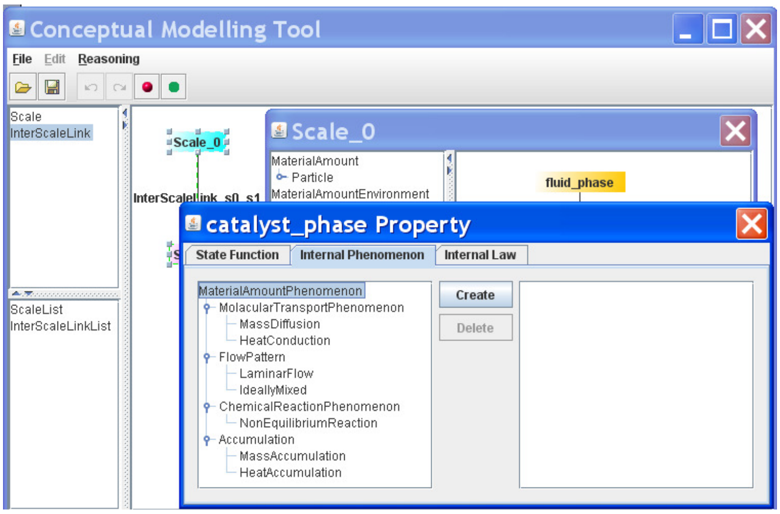


Figure 12. Specifying phenomena of a scale component.

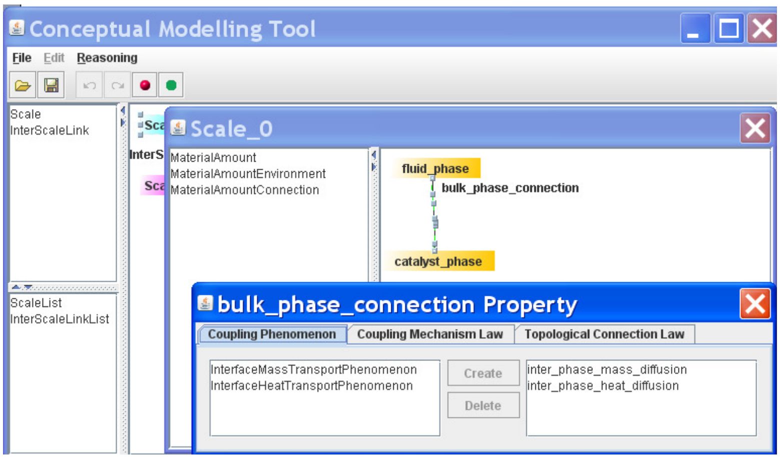
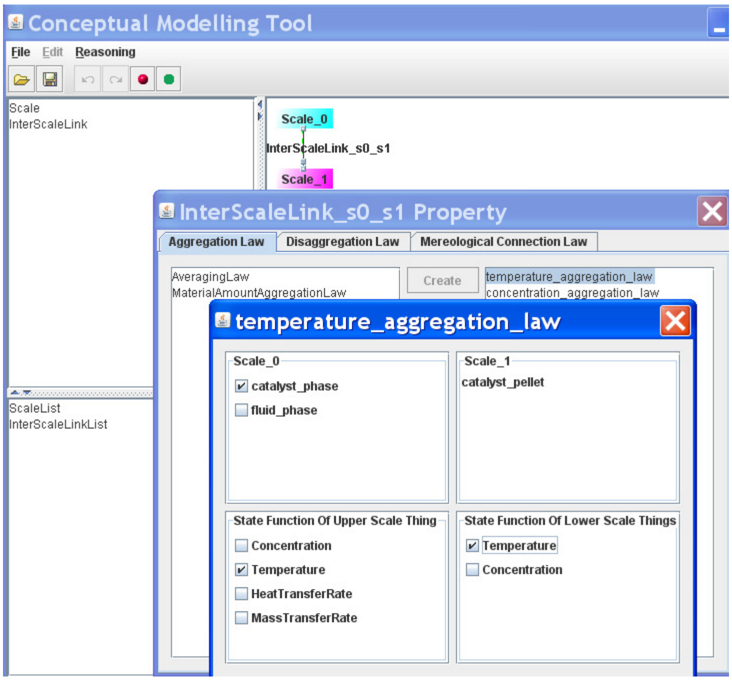

Conceptual Modelling Tool
File
Edit
Reasoning
Scale
InterScaleLink
ScaleList
InterScaleLinkList
Scale_0
MaterialAmount
MaterialAmountEnvironment
MaterialAmountConnection
fluid_phase
bulk_phase_connection
catalyst_phase
bulk_phase_connection Property
Coupling Phenomenon
Coupling Mechanism Law
Topological Connection Law
InterfaceMassTransportPhenomenon
InterfaceHeatTransportPhenomenon
Create
Delete
inter_phase_mass_diffusion
inter_phase_heat_diffusion

Figure 13. Specification of a coupling.

The above procedure can equally be applied to Scale 1, where however only one system component, namely an instance of the class *representative particle* (termed “catalyst_pellet”), is created and specified to represent the catalyst bed at a finer level of resolution (details omitted here). After both scales are specified, the modeller can now specify the interscale link. As shown in Figure 14 (particularly the window entitled “InterScaleLink_s0_s1 Property”), an interscale link can be specified in terms of *aggregation law*, *disaggregation law*, and *mereological connection law*, all as defined in the generic ontology. The left panel of the above window displays both generic aggregation laws defined in the generic ontology (e.g. *averaging law*) as well as any domain-specific aggregation laws that may be introduced through the domain ontology (e.g. *material amount aggregation law*). In this example, two instances of *averaging law* are created at the right panel. The window entitled "temperature_aggregation_law" in Figure 14 illustrates how further details of an *aggregation law* can be provided. In this case it is stated that the temperature of pellets is aggregated to obtain the temperature of the catalyst phase. *Disaggregation laws* can be specified in a similar way. This example does not involve any *mereological connection law*, therefore no instance of this concept is created.


Conceptual Modelling Tool
File
Edit
Reasoning
Scale
InterScaleLink
Scale_0
InterScaleLink_s0_s1
Scale_1
ScaleList
InterScaleLinkList
InterScaleLink_s0_s1 Property
Aggregation Law
Disaggregation Law
Mereological Connection Law
AveragingLaw
MaterialAmountAggregationLaw
Create
temperature_aggregation_law
concentration_aggregation_law
temperature_aggregation_law

Scale_0
☑ catalyst_phase
□ fluid_phase

Scale_1
catalyst_pellet

State Function Of Upper Scale Thing
□ Concentration
☑ Temperature
□ HeatTransferRate
□ MassTransferRate

State Function Of Lower Scale Things
☑ Temperature
□ Concentration

Figure 14. Specification of inter-scale laws.

4.3.1.3 Consistency checking

After the inter-scale link is fully specified, a complete conceptual model is constructed. Now the modeller may wish to ask the CCMT to check whether there is any inconsistency between the resulting conceptual model and the axioms as well as rules defined in both the generic ontology and domain ontology, particularly against the pre-selected model type ("parallel multidomain integration model" in this example as introduced in Section 4.3.1.1). The CCMT internally uses the ontology and SWRL rules reasoner to perform the checking, reporting any inconsistency it detects. As an illustration, assume the modeller has by mistake specified an internal law for heat conduction in the catalyst phase part of scale 0. As discussed earlier this would violate the definition of the model type which is selected by modellers at the beginning of the conceptual modelling stage (described in section 4.3.1.2). Checking the consistency of the conceptual model, the tool will report errors in this case, complaining that the representative pellet does not have a corresponding upper scale component that does not possess any internal law. Corrections to the conceptual model would have to be made in such a case until no inconsistency is detected by the CCMT.

4.3.2 Modelling of a homogeneous-heterogeneous reactor

In the second example, a homogeneous-heterogeneous chemical reactor (Vlachos, 1997) is used to demonstrate that the handling of a different type of multi-scale systems by the CCMT can be achieved simply by using a different domain ontology while the tool itself can remain completely unchanged. Besides, this example is intended to show a case where a modeller starts constructing the conceptual model without pre-selecting a specific model type, but he or she uses the CCMT to classify the resulting conceptual model according to the model types defined in the generic ontology.

4.3.2.1 The modelling problem

The reactor to be modelled comprises a homogeneous bulk fluid phase and a heterogeneous solid catalyst surface; the latter is to be characterised at the continuum level and additionally by a molecular-lattice. In the bulk fluid phase, the key phenomenon is the diffusion of reactants and products. Chemical reactions are modelled at a macroscopic reaction surface which is further supplemented by the modelling of micro-kinetics at the level of a molecular lattice. On each site of the lattice, (micro) adsorption, desorption or reaction may occur. As such the reactor can be modelled as a two-scale system. At scale 0, the homogeneous bulk fluid phase connects to the macroscopic, continuum reaction surface. At scale 1, a microscopic view is given to the reaction surface, in the form of molecular lattice comprising a number of sites. The link between these two scales is presented by data aggregation and disaggregation. As for the continuum reaction surface at scale 0, a state function representing the fraction of occupied surface is required for calculating the particular kinetics rate related to the lateral interactions of species at the surface. This state function is determined by aggregating the occupation-site function and the local coordination of every site of the molecular-lattice at scale 1. On the other hand, for each site of the molecular lattice at scale 1, two properties or state functions need to be characterized, namely temperature and concentration at the fluid phase boundary layer adjacent to the site. These properties of the site are determined by the corresponding properties of the continuum reaction surface at scale 0, by means of disaggregation relations.

4.3.2.2 The process of constructing the conceptual model

Same as in the previous example, the construction of the conceptual model starts with loading a domain ontology. The domain ontology used in the previous example has all the concepts required for representing the bulk fluid phase, but extension is required for the rest of the current example. This is particularly true for describing the system components at scale 1 (i.e. molecular lattice sites) and their state functions, phenomena, and laws. After the required extensions are made, the modeller can go through the same procedure as the previous example except that a model type is not chosen at the beginning, with the assumption that in this case the modeller opts to carry out the modelling without confining to a specific model type a priori. Figure 15 shows a screen snapshot at the step where the components at scale 1 are specified. In the window entitled “Scale_1”, the right panel contains a representation of the molecular lattice by means of an instance of *representative surface lattice site* (referred to as the central site) and four instances of *surface lattice site* that are connected to the former as its neighbours in the lattice. The window entitled “RepresentativeSurfaceLatticeSite_CentralSite” shows that the central site is specified in terms of state function, internal phenomenon, and internal law, in a way which is same as how every system component was specified in the previous example (see Section 4.3.1.2). What is particularly shown in the right panel of this window is a list of state function instances declared for the central site, referring to the classes shown in the left panel. In the left panel, one can see a number of state function classes newly introduced to the domain ontology for this example, including (micro event) rates, event probabilities, local coordination, etc.

4.3.2.3 Model classification

After the conceptual model is constructed, the modeller can now invoke the classification function of the CCMT. Figure 16 shows the output of the tool, indicating the resulting model (named internally by the CCMT as "ScaleIntegrationModel_Scale_0_Scale_1") is of the type "parallel embedded integration model". This is a correct result in that this model type is defined in the generic ontology as a parallel model (i.e. one with both aggregation and disaggregation of data between neighbouring scales) which involves laws modelling the same part of a system but introduced at multiple scales. This is exactly the case of the conceptual model resulting from this example: the two-way link between the molecular lattice at scale 1 and the continuum reaction surface at scale 0 makes it a parallel model, whilst the chemical reaction modelling of the catalyst surface at both the continuum and the discrete scales further gives the model an "embedded" nature. More specifically, the molecular lattice model provides critical information required by the macroscopic surface model, i.e. the fraction of occupied surface required for calculating the particular kinetics rate related to the lateral interactions of species at the surface. Therefore, the molecular lattice model can be viewed as being embedded into the macroscopic surface model. As mentioned generally in Section 4.1.2, the resulting classification may be used by the tools that support the realisation of the model following the stage of conceptual modelling. For example, the recognition of it being a *parallel* model will call for both an aggregator component and a dissaggregator component to be placed between the two models of the individual scales (cf. Zhao et al, 2011).   .

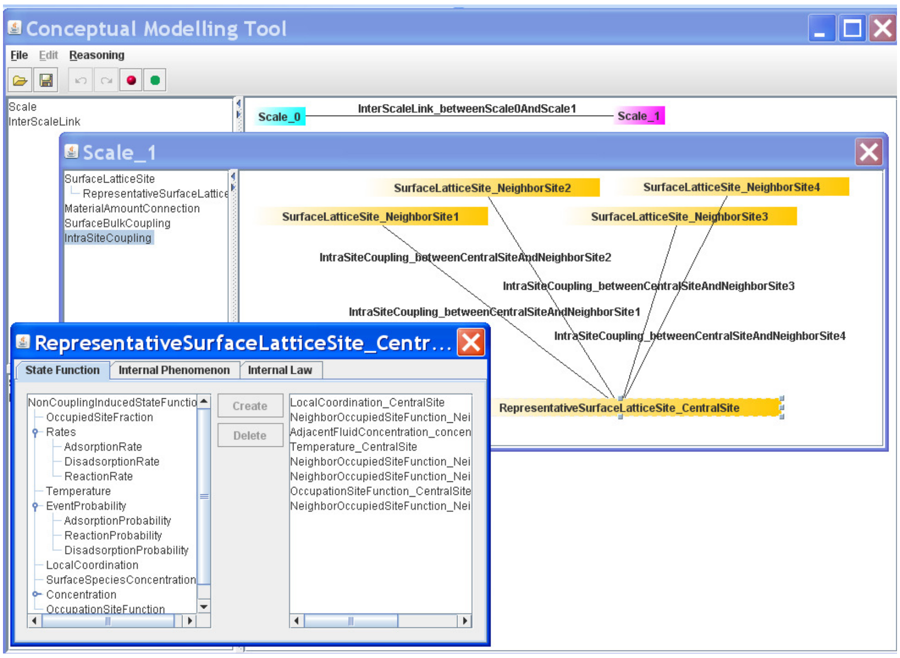
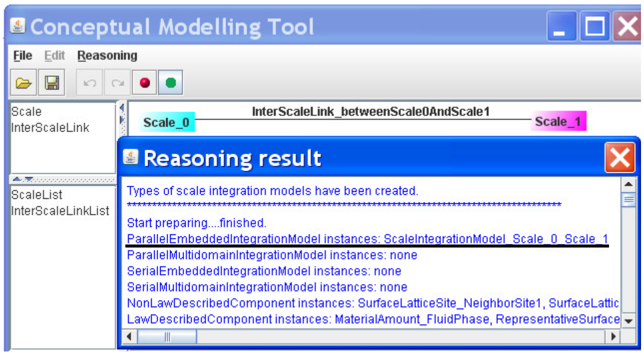

Conceptual Modelling Tool
File Edit Reasoning
Scale
InterScaleLink
Scale_0
InterScaleLink_betweenScale0AndScale1
Scale_1
Scale_1
SurfaceLatticeSite
RepresentativeSurfaceLattice
MaterialAmountConnection
SurfaceBulkCoupling
IntraSiteCoupling
SurfaceLatticeSite_NeighborSite2
SurfaceLatticeSite_NeighborSite4
SurfaceLatticeSite_NeighborSite1
SurfaceLatticeSite_NeighborSite3
IntraSiteCoupling_betweenCentralSiteAndNeighborSite2
IntraSiteCoupling_betweenCentralSiteAndNeighborSite3
IntraSiteCoupling_betweenCentralSiteAndNeighborSite1
IntraSiteCoupling_betweenCentralSiteAndNeighborSite4
RepresentativeSurfaceLatticeSite_CentralSite
RepresentativeSurfaceLatticeSite_Centr...
State Function
Internal Phenomenon
Internal Law
NonCouplingInducedStateFunctio
OccupiedSiteFraction
Rates
AdsorptionRate
DisadsorptionRate
ReactionRate
Temperature
EventProbability
AdsorptionProbability
ReactionProbability
DisadsorptionProbability
LocalCoordination
SurfaceSpeciesConcentration
Concentration
OccupationSiteFunction
Create
Delete
LocalCoordination_CentralSite
NeighborOccupiedSiteFunction_Nei
AdjacentFluidConcentration_concen
Temperature_CentralSite
NeighborOccupiedSiteFunction_Nei
NeighborOccupiedSiteFunction_Nei
OccupationSiteFunction_CentralSite
NeighborOccupiedSiteFunction_Nei

Figure 15. Specification of Scale 1 and its components.



```
Types of scale integration models have been created.
********************************************************************************
Start preparing....finished.
ParallelEmbeddedIntegrationModel instances: ScaleIntegrationModel_Scale_0_Scale_1
ParallelMultidomainIntegrationModel instances: none
SerialEmbeddedIntegrationModel instances: none
SerialMultidomainIntegrationModel instances: none
NonLawDescribedComponent instances: SurfaceLatticeSite_NeighborSite1, SurfaceLattic
LawDescribedComponent instances: MaterialAmount_FluidPhase, RepresentativeSurface
```

Figure 16. Classification of an resulting conceptual model.

5. Concluding Remarks

As a generic and promising modelling paradigm, multiscale modelling is yet to gain computer-based support in order to help human modellers overcome difficulties in conceptual, numerical, and software implementation aspects. The development of such support may be guided by a methodology which is grounded on a unified, hierarchical framework of multiscale systems theories to achieve a high degree of generality. The methodology further suggests building a set of tools around three successive modelling stages in order to maximise the facilitation the tools can offer to human modellers. This methodology has been practiced in the development of CCMT, a prototypical tool for conceptual modelling. Through two application examples, the CCMT has demonstrated the expected generality in the sense that the modelling of different types of multiscale system can be handled by taking different domain ontologies (representing domain-specific conceptualisations of multiscale systems) as input, as long as these domain ontologies are derived on the basis of the generic ontology (representing a generic conceptualisation of multiscale systems) which is also the basis on which the CCMT is built. Addressing the subsequent stages of CAMM, tools that make use of conceptual models to support model realisation and execution may be developed following the same principle to attain generality; an initial exploration in this regard is reported elsewhere (Zhao et al., 2011). Furthermore, it is envisaged that the framework of computer-aided multiscale modelling proposed in this work can be extended to support the modelling of both physical systems as well as engineering activities and procedures; extensions of the generic ontology to explicitly support discrete-event modelling are ongoing work.

Despite the progress made in this and other relevant work, it is clear that computer-aided multiscale modelling, in comparison with computer support to conventional process modelling, is still in its very early stage. The initiative reported in this paper is undoubtedly of an explorative nature; long term efforts with multidisciplinary collaborations are expected in order to make substantial progresses in this area.

Acknowledgement

This work is supported by the UK Engineering and Physical Science Research Council (EPSRC) under Grant No EP/G008361/1.

References

Batres, R., Aoyama, A., & Naka, Y. (2002). A life-cycle approach for model reuse and exchange. *Computers and Chemical Engineering*, 26, 487-498.

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

Bezzo, F., Macchietto, S., & Pantelides, C. C. (2005). Computational issues in hybrid multizonal/computational fluid dynamics models. *AIChE Journal*, 51, 1169-1177.

Bieszczad, J. (2000). *A Framework for the Language and Logic of Computer-Aided Phenomena-Based Process Modeling*. PhD Dissertation, Massachusetts Institute of Technology.

Bogusch. R., Lohmann. B., Marquardt. W. (2001). Computer-aided process modelling with MODKIT. *Computers and Chemical Engineering*, 25, 963-995.

Braatz, R. D., Alkire, R. C., Rusli, E., & Drews, T. O. (2004). Multiscale systems engineering with applications to chemical reaction processes. *Chemical Engineering Science*, 59, 5623-5628.

Braatz, R. D., Alkire, R. C., Seebauer, E., Rusli, E., Gunawan, R., Drews, T. O. (2006). Perspectives on the design and control of multiscale systems. *Journal of Process Control*, 16, 193-204.

Braunschweig, B. L., Pantelides, C. C., Britt, H. I., & Sama, S. (2000). Process modelling: The promise of open software architectures. *Chemical Engineering Progress* , 96, 65-76.

Charpentier, J. C. (2002). The triplet “molecular processes–product–process” engineering: the future of chemical engineering ? *Chemical Engineering Science*, 57, 4667-4690.

Charpentier, J. C. (2009). Perspective on multiscale methodology for product design and engineering. *Computers and Chemical Engineering*, 33,936-946.

Couenne, F., Jallut, C., Maschke, B., Breedveld, P., Tayakout, M. (2006). Bond graph modelling for chemical reactors. *Mathematical and Computer Modelling of Dynamical Systems*, 12, 159 – 174.

Couenne, F., Jallut, C., Maschke, B., Tayakout, M., Breedveld, P. (2008). Structured modeling for processes: a thermodynamical network theory. *Computers and Chemical Engineering*, 32, 1128–1142.

Doi, M. (2002). Material modelling platform. *Journal of Computational and Applied Mathematics*, 149, 13-25.

Feig, M., Karanicolas, J., & Brooks III, C. L. (2004). MMTSB Tool Set:next term enhanced sampling and multiscale modelling methods for applications in structural biology. *Journal of Molecular Graphics and Modelling*, 22, 377-395.

Fraga, E. S., Wills, G., Fairweather, M., & Perris, T. (2006). “Smart Models” - a framework for adaptive multi-scale modelling. *Computer Aided Chemical Engineering*, 21, 457-462.

Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisitions*, 5, 199-220.

Hetherington, J., Bogle, I. D. L., & Saffrey, P. (2007). Addressing the challenges of multiscale model management in systems biology. *Computers and Chemical Engineering*, 31, 962-979.

Hunter, P. J., Burrowes, K., Fernandez, J., Nielsen, P., Smith, N., & Tawhai, M. (2005). The IUPS Physiome Project: Progress and Plans. In: Kriete, A., Eils, R., editors. *Computational Systems Biology*. Oxford: Elsevier, 383-394.

Ingram, G. D., Cameron, I. T., & Hangos, K. M. (2004). Classification and analysis of integrating frameworks in multiscale modelling. *Chemical Engineering Science*, 59, 2171-2187.

Jensen, A. K., & Gani, R. (1996). A computer aided system for generation of problem specific process models. *Computers and Chemical Engineering*, 20, S145-150.

Karakasidis, T. E., & Charitidis, C. (2007). A. Multiscale modelling in nanomaterials science. *Materials Science and Engineering*, 27, 1082-1089.

Kraines, S. B., Batres, R., Koyama, M., Wallace, D. R., & Komiyama, H. (2005). Internet-based integrated environmental assessment: using ontologies to share computational models. *Journal of Industraial Ecology*, 9, 31-50.

Kulikov, V., Briesen, H., & Marquardt, W. (2005). Scale integration for the coupled simulation of crystallization and fluid dynamics. *Chemical Engineering Research and Design*, 83, 706-717.

Li, J., Ge, W., Zhang, E. J., & Kwauk, M. (2005). Multi-scale compromise and multi-level correlation in complex systems. *Chemical Engineering Research and Design*, 83, 574-582.

Liu, A., Rugonyi, S., Pentecost, J. O., Thornburg, K. L. (2007). Finite element modelling of blood flow-induced mechanical forces in the outflow tract of chick embryonic hearts. *Computers & Structures*, 85, 727-738.

Lucia, A. (2010). Multi-scale methods and complex processes: A survey and look ahead. *Computers and Chemical Engineering*, 34, 1467-1475.

Mangold, M., Motz, S., Gilles, E.D. (2002). A network theory for the structured modelling of chemical processes, *Chemical Engineering Science*, 57, 4099-4116.

Maroudas, D. (2000). Multiscale modelling of hard materials: Challenges and opportunities for chemical engineering. *AIChE Journal*, 46, 878-882.

Marquardt, W. (1995). Trends in computer-aided process modelling. *Computers and Chemical Engineering*, 20, 591-609.

Marquardt, W., Morbach, J., Wiesner, A., Yang, A. (2010). *OntoCAPE - A Re-Usable Ontology for Chemical Process Engineering*. Heidelburg: Springer.

Marquardt, W., Wedel, L. V., & Bayer, B. (2000). Perspectives on lifecycle process modelling. *AIChE Journal*, 26, 192-214.

Morbach, J., Yang, A., & Marquardt, W. (2007). OntoCAPE - a large-scale ontology for chemical process engineering. *Engineering Applications of Artificial Intelligence*, 20, 147-161.

Morales-Rodríguez, R., & Gani, R. (2007). Computer-aided multiscale modelling for chemical process engineering. *Computer Aided Chemical Engineering*, 24, 207-212.

O'Connell, S. T., & Thompson, P. A. (1995). Molecular dynamics–continuum hybrid computations: A tool for studying complex fluid flows. *Physical Review E*, 52, 5792-5795.

Pantelides, C. C. (2001). New challenges and opportunities for process modelling. *Computer Aided Chemical Engineering*, 9, 15-26.

Perkins, J. D., Sargent, R. W., Vázquez-Román, R., & Cho, J. H. (1996). Computer generation of processnext term models. *Computers and Chemical Engineering*, 20, 635-639.

Rusli, E., Drews, T. O., & Braatz, R. D. (2004). Systems analysis and design of dynamically coupled multiscale reactor simulation codes. *Chemical Engineering Science*, 59, 5607-5613.

Schopfer, G., Yang, A., Wedel, L. v., Marquardt, W. (2004). CHEOPS: A Tool-Integration Platform for Chemical Process Modelling and Simulation. *International J. on Software Tools for Technology Transfer*, 6, 186-202.

Smirnov, A. V. (2004). Multi-physics modelling environment for continuum and discrete dynamics. *International Journal of Modelling and Simulation*, 24, 190-197.

Stephanopoulos, G., Henning, G., & Leone, H. (1990). MODEL.LA: A modelling language for process engineering. *Computers and Chemical Engineering*, 14, 813-846.

Southern, J., Pitt-Francis, J., Whiteley, J., Stokeley, D., Kobashi, H., Nobes, R. (2008). Multi-scale computational modelling in biology and physiology. *Progress in Biophysics and Molecular Biology* 96, 60-89.

Vlachos, D. G. (1997). Multiscale integration hybrid algorithms for homogeneous–heterogeneous reactors. *AIChE Journal*, 43, 3031-3041.

Vlachos, D. G. (2005). A review of multiscale analysis: Examples from systems biology, materials engineering, and other fluid-surface interacting systems. *Advanves in Chemical Engineering*, 30, 1-61.

Westerweele, M. R., Preisig, H. A., & Weiss, M.(1999). Concept and design of modeller, a computer-aided modelling tool. *Computers and Chemical Engineering*, 23, S751-754.

Yang, A., Morbach, J., & Marquardt, W. (2004). From conceptualization to model generation: the roles of ontologies in process modelling. In: Floudas CA, Agrarwal R , editors. Proceedings of FOCAPD 2004; 2004 July 11-16; New Jersey, USA; New York: American Institute of Chemical Engineers, 591-594.

Yang, A., & Marquardt, W. (2009). An ontological conceptualization of multiscale models. *Computers and Chemical Engineering*, 33, 822-837.

Zhao, Y. (2010). *Computer-aided multiscale modelling.* PhD Transfer Report, University of Surrey.

Zhao, Y., Jiang, C., & Yang, A. (2010). Conceptual and numerical support to the development of multiscale models. *Computer Aided Chemical Engineering*, 28, 1667-1672.

Zhao, Y., Jiang, C., & Yang, A. (2011). Towards a generic simulation environment for multiscale modelling based on tool integration. *Computer Aided Chemical Engineering*, 29, 76-80.