

Critical evaluation of paradigms for modelling integrated supply chains

Koen H. van Dam ^{a,*}, Arief Adhitya ^b,
Rajagopalan Srinivasan ^{b,c}, Zofia Lukszo ^a

^a*Department of Energy and Industry, Faculty of Technology, Policy and Management, Delft University of Technology (TU Delft), Jaffalaan 5, 2628 BX Delft, the Netherlands*

^b*Institute of Chemical and Engineering Sciences, 1 Pesek Road, Jurong Island, Singapore 627833, Singapore*

^c*Department of Chemical and Biomolecular Engineering, National University of Singapore, 4 Engineering Drive 4, Singapore 117576, Singapore*

Abstract

Contemporary problems in process systems engineering often require model-based decision support tool. Among the various modelling paradigms, equation-based models and agent-based models are widely used to develop dynamic models of systems. Which is the most appropriate modelling paradigm for a supply chain? In this paper, we seek to address this important question through a well-structured benchmarking process. First, we demonstrate that in the space of models, 'equations' and 'agents' are concepts of a different order, the former referring to the system description elements in the model while the latter emphasises the model elements. Thus conceptually, the two paradigms are not mutually exclusive. Next, in a case study different dynamic models of an oil refinery supply chain are developed, using different tools and approaches. By performing detailed experiments with two different models, it is demonstrated that the models are equivalent when compared using model definition, numerical results and recommended decisions. However, the modelling process itself is different and results in different model structures. By analysing the effort required to expand the models, allowing new scenarios to be tested, and reuse of model components, we identify the strengths of the two paradigms in the context of supply chain modelling.

Key words: Benchmarking, modelling, agent-based models, equation-based models, oil refinery, supply chain

* Corresponding author.

Email address: K.H.vanDam@tudelft.nl (Koen H. van Dam).

1 Introduction

Contemporary problems in Process Systems Engineering (PSE) often require a model of the process, product, or system for their solution [1] [2] [3]. There are many ways to model a system depending on the problem characteristics, purpose of modelling, functional specifications, available information, etc [4]. In contrast to traditional process systems where artifacts with physico-chemical interactions are the key constituents, supply chains (SCs) are best thought of as socio-technical systems where complex production technologies interact with distributed, intelligent, autonomous entities – each with their own dynamics, goals, desires and plans.

There is a significant challenge in modelling such systems that function in dynamic, stochastic, socio-economic environments with intra-organisational and inter-organisational complexity. Numerical modelling, traditionally the paradigm of choice in PSE, could be adopted to represent such complex socio-technical systems. One subset of these, optimisation-models, have been widely used, especially when the scope of the problem is limited to selected supply chain functions such as planning [5] or scheduling [6]. Simulation models were envisaged for supply chains as far back as the 1950s [7], however it is only more recently that dynamic simulation-based decision support are attracting wide-spread attention, especially in the PSE community [8] [9] [10].

Supply chain dynamics can be modelled through balance equations (for inventory, orders, etc) similar in structure to those used to model chemical processes [11]. The socio-technical nature of supply chain problems, however, motivates an alternative modelling paradigm: agent-based models. These take an actor-centric perspective instead of the activity-based one. The actions of each actor – represented as an agent – and the interactions between them are explicitly represented in such models, and in consequence the behaviour of the entire system emerges.

Agent-based models are now widely considered to be a promising approach for decision support in supply chains [12] [13] [14] [15] [16] [17]. A detailed comparison of the advantages and disadvantages of the equation-based and agent-based modelling paradigms that can be used for supply chains is called for. By building different models and looking, for example, at how they are built and how they can be expanded, we can come to a well-founded justification of the choice of modelling paradigm, recommendations and guidelines on which paradigm is more suitable for which application or problem.

Others, including Parunak [18], Borshchev and Filippov [19] and Tang *et al.* [20], have attempted to perform a similarly motivated comparison between equation-based models (EBMs) and agent-based models (ABMs). However,

in all these papers a clear definition of what is being compared is missing. Inconsistency in the definitions of modelling paradigms has led to a situation where conclusions from one author are used as an unfair justification for the choice of a certain modelling paradigm by others.

Parunak *et al.* [18] were the first to compare ABMs, then a very new field of research, with the traditionally used EBMs. They write that “understanding the relative capabilities of these two approaches is of great ethical and practical interest to system modellers and simulators”. The case study they use is that of the (relatively simple) Forrester supply chain. In their study, the EBM uses differential equations, while the ABM is more detailed comprising different classes of agents (e.g. company agents representing the different companies in the supply chain, shipping agents used to model transport with uncertainty and delays, etc).

After performing a comparison between the models, they conclude that ABMs can be applied to all domains that traditional models have been previously used in, and that there are some advantages such as a more natural fit, ease of construction, support for more direct experiments, and the ease of translation back to practice. Perhaps some of these benefits do accrue, but good measures or indicators to establish them have been hard to come by. Even though it does not compare two models made for the same purpose, the article’s conclusions (fuelled by the fact that the term EBM used in the title is much broader than the system model examined in its body) are often used to justify the application of the agent-based paradigm.

Chatfield *et al.* [21] discussed the use of different formalisms combined, to take advantage of strong points of each: “Forcing modelers to conform their understanding of a subsystem to an unnatural viewpoint may lead to added model building difficulty. For example, agent-based concepts are easily mapped to some supply chain entities and actions, such as basic supply chain participants (retailers, warehouses, etc) and their behaviors, but are not suited for other areas of the supply chain, such as process items (materials, orders, etc.)”

Macal and North [22] [23] developed an agent-based implementation of the beer game (a frequently used case study for research on supply chains), based on the original System Dynamics model. They claim that their results “exactly duplicate” Sterman’s EBM [24] which once again demonstrates that dynamics of an EBM can be captured by an ABM. While insightful, the results from these comparisons are obtained without resorting to a well-defined approach and without clear definitions of what is compared, making it difficult to generalise the findings.

We would like to stress that comparing modelling paradigms based only on the conceptual model specifications is not enough; rather a well-defined bench-

marking process and the execution of experiments are required. In this paper, we present a strategy for executing benchmarking studies of modelling paradigms. A benchmarking exercise is executed for two models (each motivated by a different paradigm) of an oil refinery supply chain. The lessons learnt will be applicable for modelling supply chains, but can be generalised to other socio-technical systems, for example in the infrastructure domain.

The rest of this paper is structured as follows. First, in Section 2 we will discuss the differences between modelling paradigms and stress the importance of clearly defining the label used for a paradigm. The corner-stone of this paper is the detailed benchmarking process described in Section 3. After a description of the oil refinery supply chain in Section 4, three different models of this system are presented in Section 5. The benchmarking process is then applied to the refinery supply chain models and conclusions from the exercise are drawn and recommendations for the use of the different modelling paradigms are given in Section 6. Finally, in Section 7, the paper is summarised.

2 Modelling paradigm spectrum

Many researchers use different definitions of ABMs. Statements made about the advantages of the approach are too easily generalised to models that fall outside the definition. The same pitfall could be observed when the term EBM is used. This inconsistency has led to a situation where conclusions from one author are used as a justification for the choice of a certain modelling paradigm where this is not suitable.

To identify commonality among the various perspectives on agent-based modelling, a small survey was designed and sent to a group of researchers with a strong interest and contribution to the agent-based systems area (see Acknowledgements). It became clear that the concept “agent” has a different meaning when used in “agent-based models” and “multi-agent system”. When talking about multi-agent systems, characteristics following, for example, definitions by Wooldridge and Jennings [25] are common. When used in an ABM context, it appears to be more a metaphor – or way of thinking – towards modelling the behaviour of individuals, rather than a strict definition with minimum requirements.

One conclusion that we can draw is that there is not a clear line between agent-based and non-agent based models. The concept is not black-and-white, rather there is a continuous scale (or a spectrum in the modelling space) where a model can be more agent-based or less so. There are two main axes in which models can differ: The *model elements* axis and *system description elements* axis. The former deals with *what* is modelled and the constituents of the model,

Table 1

Attributes of modelling. The arrows illustrate that there is not a strict division between the rows but a continuous scale

Model label	Model elements	System description elements	Implementation platform
<p>Equation-based Model (EBM)</p> <p style="text-align: center;">⇕</p> <p>Agent-based Model (ABM)</p>	<p>System observables (flows & states)</p> <p style="text-align: center;">⇕</p> <p>Individuals (decision making entities & executing entities)</p>	<p>Equations</p> <p style="text-align: center;">⇕</p> <p>Algorithms</p>	<p>Mathematical software tools</p> <p style="text-align: center;">⇕</p> <p>Agent-based software tools</p>

the latter with *how* their structure and behaviour is formally described. After an attempt to define these concepts an illustrative example will follow.

First, we will consider *what* is being modelled. The model elements can range between *system observables* and *individuals*. System observables are the flows and states that can be observed in the real system, without taking into account who or what caused them (and why). On the other end of the spectrum, a focus on individuals means that the system is modeled by capturing the behaviours of exactly these decision making and executing entities. The behaviour of an individual leads to actions that, together with the actions of all other individuals, cause system level behaviour, which can be observed in the model.

Next, there are different ways to formalise the structure and behaviour (or in other words, *how* the model is built). Various description elements such as *equations* or *algorithms* can be used. An equation is a mathematical statement that two terms on either side of the equals sign are equivalent. Algorithms are well-defined sequences of instructions.

ABM and EBM are *labels* used to describe a model and they can be characterised by their use of these model elements and system description elements. As for ABMs, in general they are characterised by a focus on individuals as model elements. EBMs, on the other hand, focus on system observables modelled predominantly using equations. An essential point to be noted in this context is the following. Once any model has been constructed it has to be simulated or solved. The presence of algorithms in the model description is qualitatively different from those being used in the solution procedure. Both ABMs and EBMs would require algorithms in the solution procedure, however, only ABMs would contain algorithms in the model description itself.

Table 1 shows the model labels, their predominant model and system descrip-

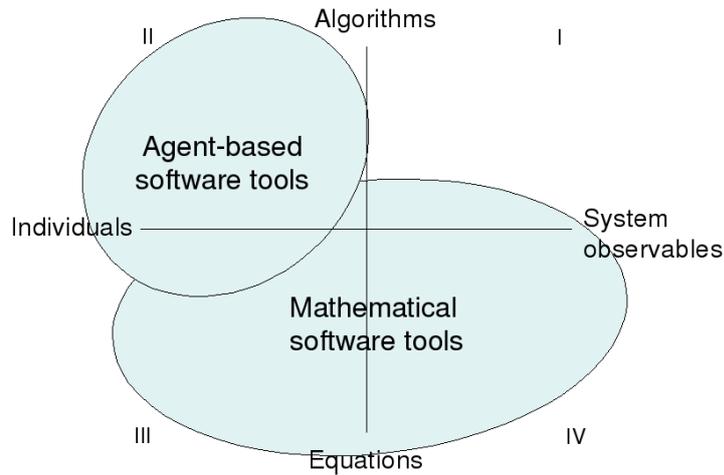


Fig. 1. The space of models based on algorithms or equations, and focus on individuals or system observables. The areas indicating which software tools are most applicable are without clear borders

tion elements, as well as and the commonly used implementation platform, for both EBMs and ABMs. There is no strict division between the rows. Equations, for example, are system description elements that can be used to describe certain effects or observed behaviour, but are not exclusive to non-agent based systems. Even though they may be predominant in models built up from system observables, they may be applied in individual-based models too.

Where ABMs are mostly identified by the model elements (second column), EBMs are mostly identified by the system description elements (third column), resulting in a space in which it is not clear what label to give a model. This also means that the use of equations is not the opposite to an ABM nor is it an alternative per se, as is often stated. Rather, *agent* and *equation* are concepts of a different order.

Figure 1 illustrates the modelling space and plots possible implementation platforms that can be used to create the models. Agent-based software tools are mostly useful for individual-based models that do not use many equations, while mathematical tools are mostly applicable to any type of model that uses equations (but most can capture algorithms too).

A point *on* the x-axis illustrates that both equations and objects are used approximately to the same extent and are equally important. A point *on* the y-axis means that individuals and system level observables are of equally predominant. A point on the extremes means the model only uses one type of model or language elements. Other points in the space highlight the predom-

inant, but not exclusive, characteristic.

In general, what is called an ABM can be found in quadrant II and traditionally EBMs are in quadrant IV, but this is not exclusive as will be demonstrated in Section 5. Other examples of models in quadrant III could be purely continuous physical systems such as liquid flow or molecular dynamics where models describe the behaviour of a large number of system constituents (individual molecules) using equations. Quadrant I in which models would use algorithms to model system observables) appears to be an uncommon modelling style.

Having emphasised that there is no “black” or “white” when it comes to the label *agent-based model* or *equation-based model*, we can map models to indicate their essential characteristics. This illustrates not only how the various models are different but also to what extent they are similar. This formulation, by acknowledging the absence of a clear dichotomy, makes stark contrasts more difficult, but, for a fair benchmarking, we believe the similarities between models should also be fully captured. Note that this distinction as presented here is valid for the domain used in this paper, that of a supply chain. In other fields of study other differences between the modelling paradigms may also be prevalent.

In the next section, a benchmarking exercise will be introduced before presenting a case study (Section 4) and different models of this system (Section 5). The models can then be placed in Figure 1 to show their differences and similarities.

3 Benchmarking

Benchmarking is about making comparisons and, through these, learning generalisable lessons. It is not possible to compare modelling paradigms based only on the conceptual model specifications; rather a well-defined benchmarking process is required.

In order to assess the performance of the two modelling paradigms, the following scheme, inspired by [26] and refined in [27] and [28], is adapted:

- (1) Definition of the objectives for the study
- (2) Identification of what is to be benchmarked
- (3) Evaluation if objects of study are comparable
- (4) Determination and specification of performance measures
- (5) Description of scenarios (well-structured experiments) and their simulation
- (6) Conclusions

Next, we will explain each of these steps.

3.1 *Definition of the objectives for the study*

In the first step of the benchmarking process, the objective for the study has to be defined. Examples of suitable objectives are:

- Choosing the fastest model for online decision making support
- Learning the advantages of one modelling paradigm over another
- Justification of choice of modelling paradigm for a new project
- Testing a new modelling platform with a traditionally used or well validated approach

A clear definition enables selection of appropriate performance measures in step 4.

3.2 *Identification of what is to be benchmarked*

Next, the objects of the study should be identified. Or in other words: what is going to be benchmarked. This has to be clear and detailed so others are able to reproduce the experiments, if desired. This adds to the transparency of the benchmarking study. The objects of study should be specific models that have been implemented and that can be used to perform experiments. This step can for example refer to a detailed model description or even include the source code. It is assumed that the objects of study are comparable to make a successful benchmarking study, something that is evaluated in the next step.

3.3 *Evaluation if objects of study are comparable*

Valid and useful conclusions can only be drawn from a benchmarking study when it has been demonstrated that the objects, defined in the previous step, are actually comparable. It is not easy to say when models are comparable and even more complicated to say that they are *equal*. Perhaps it is better to talk about models being similar. But what is similar enough to justify saying that they are comparable? Inspired by Sterman [29], distinguishing between model verification and model validation, we compare models at different levels:

- (1) The definition of the models that produce the results (*model verification*)
- (2) The numerical results of the model for (selected) output variables (*result validation*)

(3) The *decision* that follows from studying the results of the models

The first step, *model verification*, deals with the way the models are formalised using equations and algorithms. It can be checked whether the model has been coded correctly and consistently. For all models the model definitions can be compared and it can be demonstrated that, even when a different *language* is used for the system description, models encode the same *behaviour*. This stage of testing has to some extent overlap with the specification of the model.

Next, in the *result validation phase*, the model behaviour is studied by comparing the numerical results of the model (e.g. for extreme conditions or other pre-defined conditions). When the definition of models is exactly the same, one can assume that if there are no model-dependent errors, then the same numerical results can be achieved as well. However, it can happen that the model definitions are precisely the same, but due to numerical-method-dependent differences (e.g. different solution methods for solving differential equations or different random numbers in the case of stochastic models) some dissimilarity in the numerical results can still be observed. The results can be exactly the same, or differ within accepted boundaries. When the numbers are the same, or close enough (e.g. only very small constant shift, small phase or amplitude difference; again, this has to be well defined), we can consider the output of the model run to be the same for that situation. Of course, that does not mean the model will behave the same in other situations, too. Furthermore, models can be deterministic (meaning that every run of the model will produce the same outcomes) or stochastic (in which “chance” or random elements are introduced so that the model output is not always equal to the output of the previous run). For stochastic models even when multiple runs of the models are averaged out, it still cannot be concluded that the output will be the same as well. But after looking at the definition of the models and how the results are produced, it is possible to get a better understanding of how models will behave under different inputs and different scenarios.

The last step compares the *decision* made using the models. Building models and running simulations is not an end by itself, but rather the means to reach a decision (e.g. policy recommendation). The values for the output variables can lead to a decision or recommendation (dependent on the goal of the model) and these can be compared. For example, do the models recommend the same policy to be deployed (qualitative similarity), even if they predict different profits based on them (quantitative similarity)? If this is the case, the models can be said to be comparable. The same problem can be observed as with the numerical results: if the outcomes are the same for one or more scenarios, it does not guarantee they will be the same for other scenarios too. When the simulation, used as a decision support tool, leads to the same decision made based on both models for one case, there is no immediate guarantee that this will also happen in a different scenario. Repeating this exercise for many

scenarios and with varied data sets gives more confidence in the similarities, but still does not *prove* that the models produce the same outcomes.

Proving that the same numerical results and the same recommendations will be produced in every scenario can only be done by comparing the definition of the models and concluding that they are the same. From this it can be deduced that the output will be equal, too. However, it should be stressed that this is not obligatory for getting useful results from a benchmarking study: when the models are comparable for certain scenarios, the benchmarking conclusions will still be useful and applicable for these scenarios. These results can still be generalised as a hypothesis that has to be tested for other scenarios.

3.4 Determination and specification of performance measures

To determine the performance measure for the different ways of modelling that are being compared it is essential to, besides a comparison of the outcomes, also reflect on the modelling exercise as a whole. For instance, it is now widely accepted that the ease of developing the model and maintaining it over the lifespan of the application is an important (sometimes critical) determinant in successful industrial acceptance [1]. Therefore, in addition to comparing the numerical simulation results from the two models, we also look at other qualitative key performance indicators.

Cavalieri [30] describes a benchmarking service for different users of control systems (e.g. researchers, vendors as well as practitioners from the industry) and performance is evaluated in terms of efficiency, robustness and flexibility. We use the same indicators here. Considering efficiency, we will look at the ease of expressing the problem in each modelling paradigm. For robustness the possibility of extending the models can be compared and for flexibility their re-usability. Inspired by the work of Cavalieri [30], an additional performance indicator is formulated: the ease of explaining the model and its applicability.

In the following sections these performance measures are discussed in more detail.

3.4.1 Ease of expressing the problem

The first step while creating a model is making a conceptual specification of the system. Subsequently, it can be operationalised by implementing it using one or more software tools, resulting in a computational model that can be used for simulations and experiments. The ease of conceptualisation of a system into a model through a paradigm is problem-dependent. This could be measured in amount of time spent in expressing the problem from conceptual design

to implementation, but one should also consider if all aspects can faithfully be included. Some aspects can be quickly implemented in one approach, but others may have to be left out or simplistic assumptions made. It is much more difficult to assign an exact measure for this; the ease of expressing the problem is therefore mostly subjective.

3.4.2 Ease of extending the models

The ease of extending the model can also be considered as a subjective measure, but much less so than most of the other performance measures. For a number of possible or desired extensions to the model (such as adding functionality to the model) a prediction can be made of the tasks that are required to bring them into being. For each of these tasks, a judgement can be made on how difficult, time consuming, and risky (to the functioning of other parts of the model) these extensions might be. The following are typical indicators that could be used (note that some may be more difficult to objectively compare than others and factors such as the experience of the modeller have to be taken into account too):

- Number of changes required
- Expected amount of time needed to complete the tasks
- Number of locations where changes have to be made
- The chance that something goes wrong
- Difficulty level of the tasks

3.4.3 Ease of re-use

An important aspect of model development, and software development in general, is the re-use of parts that have been developed previously. This could save time and resources, because earlier work can be integrated in a new project. For models, however, there is an even more important aspect to re-use: it allows models to be built upon previously validated components. When parts of models are used in different case studies and they are validated in each of them, it increases the modeler's confidence in their reliability. Furthermore, when parts of a model can be re-used it becomes easier to create larger models to study larger and more complex systems as well as interactions between different systems.

3.4.4 Ease of explaining

When used as a decision support tool one of the most important aspects, in addition to the validity of the model, is how easy it is to explain the model and interpret the results. As stated by [31] “a fundamental challenge in sim-

ulation modelling of manufacturing systems is to produce models that can be understood by the problem owner”.

Especially when the model-builder is not the same person who makes a decision based on its results, it is critical that the model outcomes can be explained so the right interpretation can be made. It can be dangerous to simply trust the model output and focus on the numbers and predictions. Instead models should be used to largely gain insight about *possible* future states as well as possible effects of actions. Again, it is not possible to put a single quantitative indicator to the easy of explaining – it has to be analysed by testing with users and their innate preferences for different descriptions (e.g. equations, algorithms or diagrams)

3.5 Description of scenarios and simulation

A benchmarking study is based on a number of scenarios that can be executed with all objects of the study. How this is done is completely case specific, but a reproducible description should always be given. Such a description might include what will be measured and which settings are used. If the objects that will be benchmarked are comparable, then the same scenarios should be used in all objects of study. Once the performance measures have been specified, the last step is to perform the simulation and evaluate the results, with the aim to learn from this benchmarking exercise and to produce a number of recommendations about the applicability and use of the approach, based on the key performance indicators (Section 3.4).

4 Case study: Oil refinery supply chain

In this paper, we look at the model of an oil refinery supply chain as a case study. It is suitable for the purpose of this paper as it comprises complex interactions among a number of decision-making actors and physical processing equipment.

The oil refinery supply chain begins from the oil reservoirs, both onshore and offshore. Crude oil is tapped from these sites and then transported to various refineries around the world mostly by pipelines or large ships called very large crude carriers (VLCCs). Transportation times of crude are relatively long; it takes four to six weeks for a VLCC carrying crude oil from the Middle East to reach refineries in Asia, for example.

The crudes are then processed in crude distillation units (CDUs) and sepa-

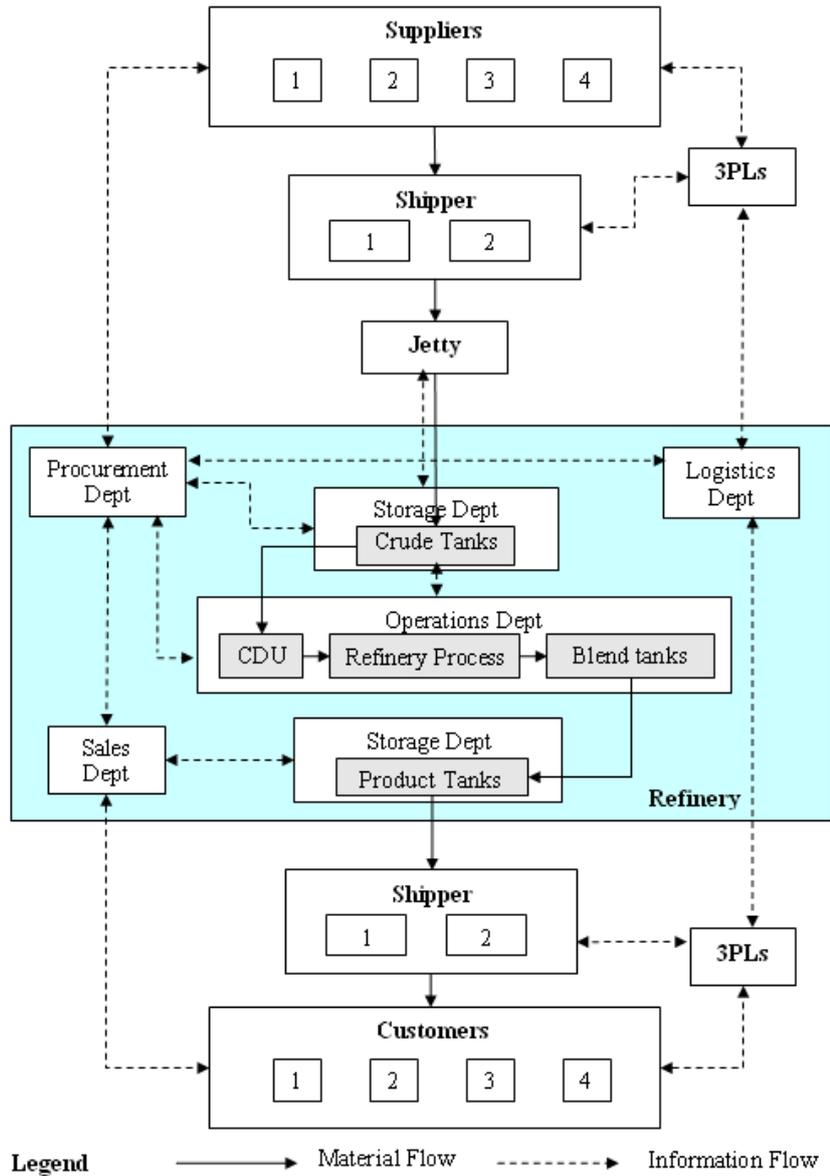


Fig. 2. Schematic of refinery supply chain (from [32])

rated into fractions based on their boiling points. These fractions are processed further in different downstream refining units such as reformer, cracker, and blending pool to get the various products. A single crude mix may yield numerous products and their variants through a suitable alteration of processing conditions. Hence, refineries must adapt their operations to the different crude batches to meet the required product specifications from their customers.

The refinery occupies a pivotal position in the supply chain with its functional departments initiating and controlling the interactions with the external entities, which are oil suppliers, 3rd party logistics (3PL) providers, shippers, jetty operators, and customers. The operation of the refinery supply chain requires

various decisions in every cycle – what mix of products to make, which crudes to purchase and in what quantities, which mix to process and in which processing mode, etc. Different actors are responsible for the different decisions [14]. These actors and their interactions are shown in Figure 2. The entities (shown as blocks in Figure 2) communicate with each other through information flows (broken arrows) in order to control the material flows (solid arrows). The refinery physical units (shaded blocks) may be further sub-divided into storage units such as crude and product tanks and processing units such as CDU, reformer, cracker, and blend tanks. The functioning of these units and other supply chain activities are overseen by the functional departments.

Each entity acts based on its policies and the combined actions of the entities determine the overall performance and economics of the supply chain. For example, the procurement department decides the type and amount of crude to buy, the logistics department oversees transportation of the crude, and the storage department manages the crude unloading from the ship to the storage tanks. The combined actions from these three departments determine crude arrival at the refinery. The complex maze of flows among the entities results in complex dynamics, which could lead to unforeseen domino effects. Furthermore, the refinery has to contend with various uncertainties such as prices, supply availability, production yields, and demand variations.

Amidst these challenges, a hierarchy of decisions has to be made in managing the supply chain: strategic (e.g. capacity investments, adding units, upgrading technology, supply chain reconfiguration), tactical (e.g. production planning, policy evaluation, disruption management), and operational (e.g. procurement, storage, scheduling, throughput level). These motivate the development of simulation models of the supply chain, which could reflect the dynamic behaviour of the entities in the face of the various uncertainties. These models enable decision making for supply chain management by allowing the user to evaluate the impact of a particular decision on the supply chain performance, analyse different supply chain policies, and identify the consequences of a disruption, through simulation.

There are different options available when choosing an appropriate modelling paradigm for supply chains. A well informed decision has to be made so the best paradigm, given the desired use of the simulation model, can be applied. Traditionally, supply chains have been modelled with equation-based models [29] but more recently the agent-based paradigm has received much attention in this field [33].

We choose to study the crude stock levels as key performance indicator of the supply chain because they represent the cumulative-effect of the various entities and their decisions. If the inventory profile shows that the maximum crude capacity is reached, it means that too much crude is being purchased

ahead of production or that procurement takes place too early, while empty storage tanks indicate that the refinery is not able to run in the desired (i.e. planned) production mode and desired throughput. Additionally, the amount of crude in storage represents an important factor for the refinery economic performance. Other options for model outputs that can be used in the future are, for example, net profit of the refinery and customer satisfaction.

Next, in Section 5, three different models of this case study will be presented.

5 Models of the oil refinery supply chain

In this section three models of the supply chain from Section 4 are described, each occupying a different place in the modelling space shown in Figure 1.

The first model is a high-level low-resolution model that can deal with throughput related equations (Section 5.1). It describes mass flows and economics of the supply chain, but lacks the elements to describe the decision making involved in the supply chain operation, for example, crude procurement. To be able to add more detail to the model one needs to consider an “individual” level description. This is done in the second model (Section 5.2). This can be considered as moving left on the horizontal axis in Figure 1 from system observables to individuals. The decisions that have to be modelled then *require* the use of algorithms, for example for the selection of which crude to buy and at which production mode to operate the process. This consequentially means moving up along the vertical system description axis, from equation to algorithm. The logic involved in transferring crude between the VLCC and the storage tanks, through the jetty, can be best described through algorithms, concentrating on the actor that *causes* each transaction (Section 5.3). After the three models have been presented, the distances between the models can be visualised in the modelling space.

5.1 Model E: An equation-based model in Excel

The first model considers only system observables and not individuals, using *only equations*. This set of equations can be implemented in a simple spreadsheet such as Microsoft Excel (hence Model E). This model is at the bottom right of the modelling spectrum, i.e. quadrant IV (See Figure 1). Modelling of decisions require algorithms, but algorithms are not used in this model. For example, “if-then” algorithm is employed in a reorder point procurement decision: if crude inventory is less than x , then order y amount of crude z . Decision making entities are thus not captured in this equations-only model.

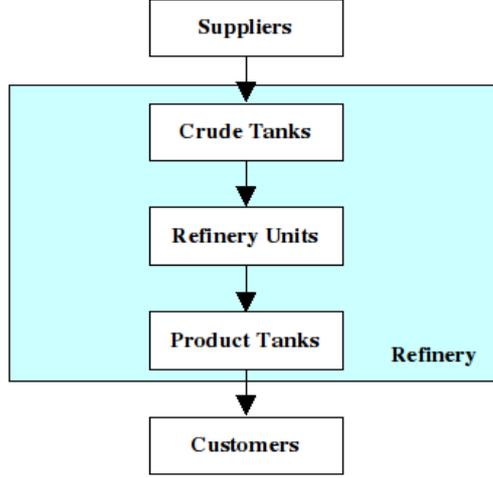


Fig. 3. Simplified schematic of refinery supply chain as used in Model E

Furthermore, as information flows mainly serve to control material flows in the form of decisions, they are not modelled. As a result, the refinery supply chain considered in this model has to be simplified, as shown in Figure 3 (c.f. Figure 2). The refinery receives crudes from the suppliers and stores them in the crude tanks. The crudes are processed in the processing units and converted to valuable products stored in the product tanks. Finally, the products are delivered to the customers.

In this model, the supply chain is modelled through equations involving system variables. Let IC_{lc} be the inventory of crude c at the beginning of cycle l , CA_{lc} the amount of crude c arriving at the refinery from the suppliers in cycle l , and TP_{lc} the amount of crude c processed in cycle l . Equation 1 describes the mass balance around the crude tanks:

$$IC_{(l+1)c} = IC_{lc} + CA_{lc} - TP_{lc} \quad (1)$$

As crude procurement, transportation, and unloading are not explicitly modelled, their effects are captured in the parameter CA_{lc} , which is an input to the model. Similarly, the decision on production throughput requires algorithms and is not modelled, thus TP_{lc} is another model input.

Conversion of crudes to products in the refinery units are modelled through simple yield calculation:

$$R_{lp} = \sum_c Y_{cp} TP_{lc} \quad (2)$$

where R_{lp} is the amount of product p produced in cycle l and Y_{cp} is the yield of product p from crude c . The processing units can be broken down further into the different units (CDU, reformer, cracker, blending) similar to [32] since they involve only equations and no algorithms. For simplicity, here they are lumped into an overall crude-to-product yield Y_{cp} .

The inventory of product p at the beginning of cycle l , IP_{lp} is obtained by mass balance:

$$IP_{(l+1)p} = IP_{lp} + R_{lp} - D_{lp} \quad (3)$$

where D_{lp} is the amount of product p delivered to customers in cycle l . This parameter D_{lp} is also an input to the model.

Besides the crude inventories, other model outputs are profit and customer satisfaction index. Profit is obtained by deducting crude procurement cost, crude inventory cost, operation cost, and product inventory cost from revenue:

$$\begin{aligned} Profit = \sum_l (\sum_p P_p D_{lp} - \sum_c P_c C A_{lc} - \sum_c InvCost_c I C_{lc} \\ - CostOp \sum_c T P_{lc} - \sum_p InvCost_p I P_{lp}) \end{aligned} \quad (4)$$

where P_p is the price of product p , P_c is the price of crude c , $InvCost_c$ is the inventory cost of crude c , $CostOp$ is the refinery operation cost, and $InvCost_p$ is the inventory cost of product p .

Customer satisfaction for product p in cycle l is measured by the ratio of product delivered to customer demand:

$$CustomerSatisfaction_{lp} = \frac{D_{lp}}{AD_{lp}} \quad (5)$$

where AD_{lp} is the actual customer demand in cycle l for product p .

Since decision making is not modelled, Model E provides limited decision support capability. Its application is limited to estimating crude inventories, profit and customer satisfaction given a set of input parameters, including crude arriving $C A_{lc}$, crude processed $T P_{lc}$ (throughput rate at which crude is sent to the CDU for processing), and product delivered D_{lp} . The next model, on the other hand, employs algorithms to explicitly capture decision making holistically within the supply chain model.

5.2 Model M: A numerical model in Matlab

The second model is a numerical model implemented in MATLAB/Simulink [34] (hence Model M). Moving left from Model E on the model element axis (Figure 1), this model has a focus on individuals – decision making entities and executing entities, which in turn necessitates the use of algorithms for describing behaviour. Four types of entities are incorporated in the model: external SC entities (e.g. suppliers), refinery functional departments (e.g. procurement), refinery units (e.g. crude distillation), and refinery economics. Some

of these entities, such as the refinery units, operate continuously while others embody discrete events, such as arrival of a VLCC. Both are considered using a unified discrete-time representation. As such it covers the full complexity of the supply chain as shown in Figure 2.

System variables – material, information, finance – are all modelled as flows interconnected by various (mathematical, logical, algorithmical) operation blocks. For better organisation and presentation, the flows and operation blocks related to a particular entity are grouped together under a masked block [35]. Hence, we have a “Supplier” masked block, a “Procurement” masked block, a “CDU” masked block, etc. In addition to these, decision making policies (procurement, unloading, production) are coded in MATLAB m-files [32]. Thus, Model M can be placed somewhere near the border between quadrant II and III (Figure 1).

Since Model M allows the modelling of crude procurement, transportation and unloading, the crude inventory balance equivalent to Equation 1 here is:

$$IC_c(t + 1) = IC_c(t) + PR_c(t) - TP_{lc} \quad (6)$$

where $PR_c(t)$ is the rate at which crude c is pumped into storage from the pipeline at time t .

Owing to the greater modelling detail, a finer simulation time step can be used: time t instead of cycle l . One time tick could be one-hundredth of a day while a cycle could have a duration of seven days. In contrast to one overall crude arrival term CA_{lc} in Equation 1, Model M has different variables for crude ordered, crude transported, and crude unloaded and pumped into storage tanks. These are not user input, but calculated by the model as an outcome of the various policies. In effect therefore, it is the policies that are input to the model.

Equation 6 is *implemented* under the “Crude Tank” masked block (See Figure 4). The main operation blocks used are an addition block (A in Figure 4), an integrator block (B), a product (i.e. multiplication) block (C), a switch block (D), and a lookup table block (E). The first three blocks embody equations. The addition block is adding the crude input flow $PR_c(t)$ and subtracting the crude output flow TP_{lc} to get the net crude movement. The integrator block calculates $IC_c(t + 1)$ from $IC_c(t)$ and the net movement. The product block multiplies the crude ratio (based on the recipe) to the crude inventory to get the correct crude mix in the throughput to CDU.

The other two blocks represent algorithms. The switch block is used to enforce a logical constraint and set a particular crude’s output to zero when its inventory reaches zero. The lookup table block is used to get the crude recipe for a production mode. Connection tags are used to convey the information of

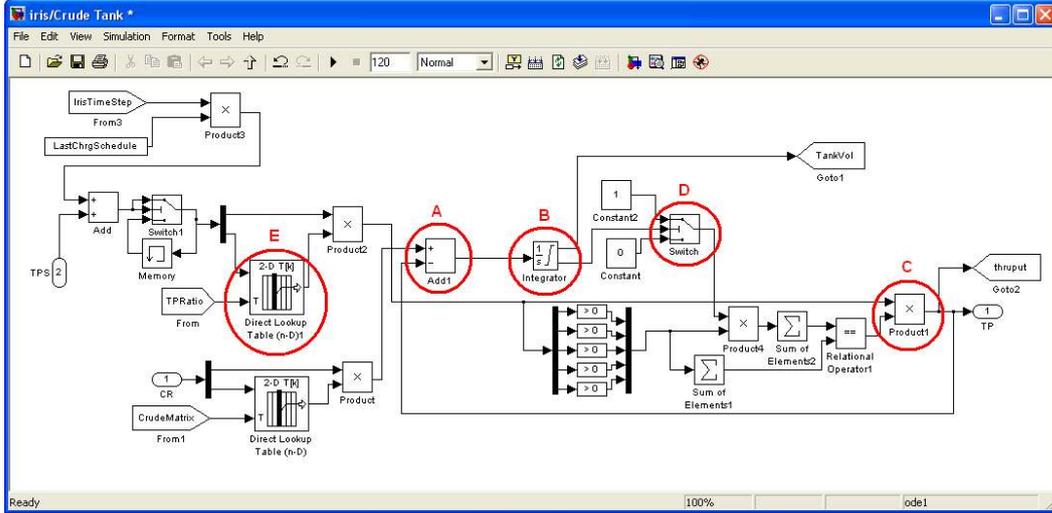


Fig. 4. “Crude tank” masked block in Model M

the flows to other masked blocks. The assigned production mode comes from the production policy. The crude input flow $PR_c(t)$ comes from the “Pipeline” masked block. The crude output flow TP_c is sent to the “CDU” masked block.

Model M is more detailed than Model E and can provide more extensive decision support. It can be used to evaluate strategic, tactical, and operational decisions, analyse different policies, and study disruption management as demonstrated in [36].

5.3 Model R: An agent-based model in Repast

A third model, using the agent-based paradigm, is created with the same level of detail as in Model M. This agent-based model is implemented in Java using the Repast agent simulation toolkit [37], hence the label Model R. The model is based on the system description from [32], using the same assumptions and with the same level of granularity. The model implementation uses a framework based on a generic ontology for socio-technical systems [38] and Java building blocks that have previously been used to model other infrastructure case studies, such as intermodal freight transport [39], CO₂ emission trading [40], and industrial clusters [41].

To construct Model R, first the agents in the system are identified: these are the actors in the supply chain, including the refinery departments (see Figure 2). Decision making rules of these actors are created, using the flows of information and data as listed above and as used in Model M. Following the ABM paradigm, the tasks are clearly distributed between the agents based on ownership. Some tasks therefore have to be split into several subtasks (requiring communication between the agents). Alongside this *social layer*,

the technical components (e.g. refinery units, jetty, etc) and their links (e.g. pipeline between the jetty and the crude storage tank) are identified. Agents then own and/or control these physical elements and make decisions about them.

All the instances of the model components, including the agents and all technical components have been stored in a knowledge base, which can be changed using a simple Graphical User Interface (GUI) provided by the Protégé tool [42] without having to adjust the model source code. The source code of the model works independently from parameters set in the knowledge base.

The ontology contains abstract descriptions of, among other things, the social and physical components, their links and properties as well as specific instances which refer to the real-life objects modelled here [38]. For physical components the possible in and out flows are defined, along with certain other properties such as maximum capacity. Furthermore, the ontology contains concepts such as ‘transport contract’ and ‘physical flow’ which are instantiated during the model run and used for the transfer of mass, information, and finance. All these elements are considered as *objects* in the Java implementation and the behaviour of agents is formalised using algorithms.

The agents in the model all act autonomously according to their own tasks. A schedule is made so that some processes (e.g. procurement) only occur at certain intervals while others (e.g. production) happen at each time step of the simulation. Events such as the arrival of a VLCC at the jetty are monitored each time step.

As an example, let us consider the agent-based implementation of Equation 6. In the agent-based model these calculations are done by storage department agent. The storage department can receive incoming crude (pumped from the jetty) and it can receive requests to release a certain amount of crude (from the operations department).

The storage department monitors the incoming flows at each time tick and sets the new level of the storage tank:

Algorithm 1 Monitoring inflow of crude (Storage Department)

```

Volume addedVolume = ((Volume) storageInFlow .
    getPhysicalProperties () . get (Volume . class ) ) ;

currentVolume . setValue ( currentVolume . getValue () +
    addedVolume . getValue () ) ;

```

where ‘addedVolume’ is the amount added to a storage tank, ‘storageInFlow’ the flow of crude from the jetty to the storage tank and ‘currentVolume’ the

current volume of crude in the tank. The first line reads the volume that is transferred into the storage and the second line sets the new value of the volume in the storage tank by adding the added amount to the current amount. It is important to note that the volume of the ‘storageInFlow’ is set by the jetty owner in another algorithm.

The operations department decides on the amount to be released based on the current production mode and current throughput of the refinery, again using an algorithm. When the amount is determined, the storage department is asked to release this from the storage tank. When the storage department receives a request to release a certain crude (from the operations department) it will first check if there is enough in stock and then create a flow to the CDU. The amount of crude in the storage tank is thus adjusted in the following way:

Algorithm 2 Release of crude (Storage Department)

```

Volume subtractedVolume = ((Volume) outFlow .
    getPhysicalProperties () . get (Volume . class ) ) ;

currentVolume . setValue ( currentVolume . getValue () -
    subtractedVolume . getValue () ) ;

```

where ‘subtractedVolume’ is the volume of crude that will be released and ‘outFlow’ the flow from the storage tank to the CDU.

All other behaviours in the model are also split up in similar fashion between the different agents. Another example of this is the selection of the production mode by the operations department, which is based on the forecasts made by the sales department and the crudes selected by the procurement department. The behaviour of the system is distributed by the division of tasks and this is implemented in a distributed fashion too in Model R.

Since Model R has the same level of granularity as Model M, it can also be used to evaluate strategic, tactical, and operational decisions, analyse different policies, and study disruption management.

5.4 *Mapping the models on the modelling space*

Three different models of the same supply chain have been presented here. The first model, Model E, only uses equations and the second model, Model M, adds algorithms for the various decisions that have to be made and that cannot be captured by equations only. The third model, Model R, implements the same behaviour as Model M, but using a different paradigm in which equations are not explicit and in which decision algorithms as well as mass

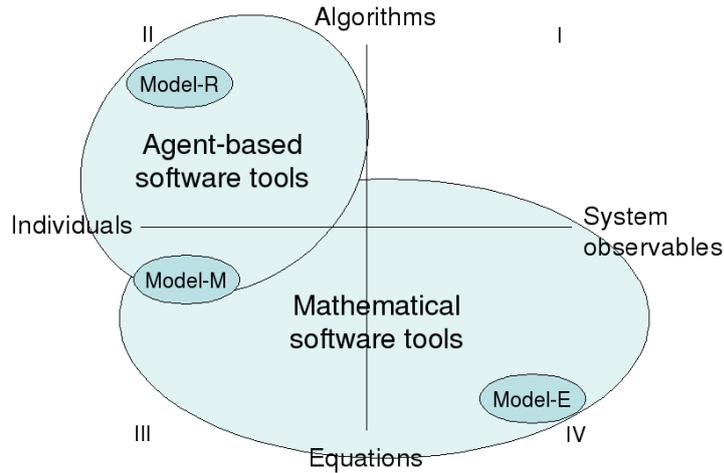


Fig. 5. The three models presented in this paper, plotted on the modelling space from Figure 1

balance equations are distributed among the actors.

While they all have different designs, there are also many similarities between the models (see Figure 5). Model E and Model M are both implemented in mathematical software tools and equations are the predominant system description elements. As such both could be labelled *equation-based models*. However, these two models are very different with respect to their model elements: Model E includes system observables only while Model M takes individuals as the constituents of the model. This brings Model M closer to Model R than to Model E. The fact that Model M and E share the same category of software tools can hide the differences between them, but Figure 5 reveals that they do not have much else in common.

Model E is included in this study to demonstrate that the same system can broadly be described using only equations, but that some of the behaviour cannot be captured without algorithms. Profit and customer satisfaction are output variables of the modelled actions of individuals in Models M and R these, while in Model E they are observed in the real system and have to be provided as input by the user. Because the scope of Model E is different, we focus only on models M and R in the benchmarking study. Still, it should be stressed that Model E is an important category to include here to illustrate a class of models that is often used when comparing equation-based and agent-based models and to highlight that conclusions drawn from such a comparison are not valid.

6 Benchmarking case study: Oil refinery supply chain

Here we benchmark the two models of the oil refinery supply chain following the steps outlined in Section 3.

6.1 *Definition of the objectives for the study*

The objective is discovering the added value of different modelling paradigms (not models) for supply chain management.

6.2 *Identification of what is to be benchmarked*

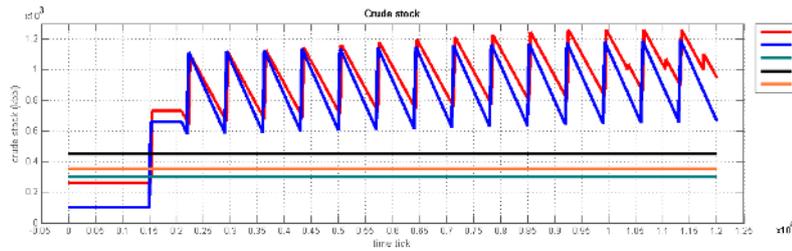
The objects of the study are two modelling paradigms that produced two different models of an oil refinery supply chain, presented in Section 5: Model M (Section 5.2) and Model R (Section 5.3). As discussed in Section 5.4, they differ along different axis. Model E (Section 5.1) is not included in this benchmarking study because conclusions drawn from it would not be fair in comparison with the other two models given the differences in the granularity.

6.3 *Evaluation if objects of study are comparable*

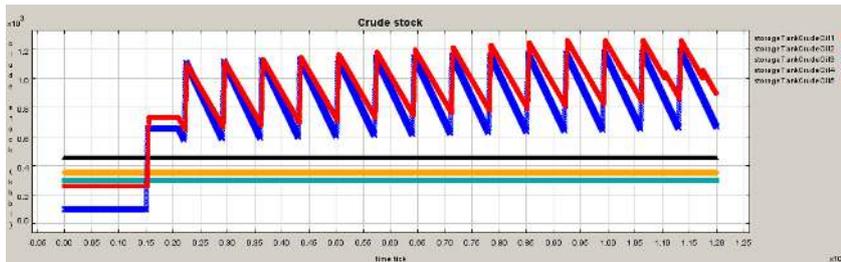
As said in Section 3.3, there are three different levels at which it can be demonstrated that the objects of study are comparable. The following sections address the model verification, result validation and policy recommendations.

6.3.1 *Model verification*

Both models are computational models and – even though the behaviour is formulated with different system description elements – the actual calculations are exactly the same. As an example of this let us look at the calculations for the amount of crude in stock at a given moment in time as formulated by Equation 6. In Model M this is implemented using masked blocks, and in Model R these calculations are done by storage department agent using addition functions in Java (Algorithms 1 and 2) of which the value is determined by algorithms in other agents. The same comparison can be made for other equations, and each time the same calculations are made but the formalisation of the calculation is done differently.



(a) Model M



(b) Model R

Fig. 6. Results from Case 1: no stochastics. Time (in simulation ticks) is on the horizontal axis and the inventory level (in kbbbl) is on the vertical axis.

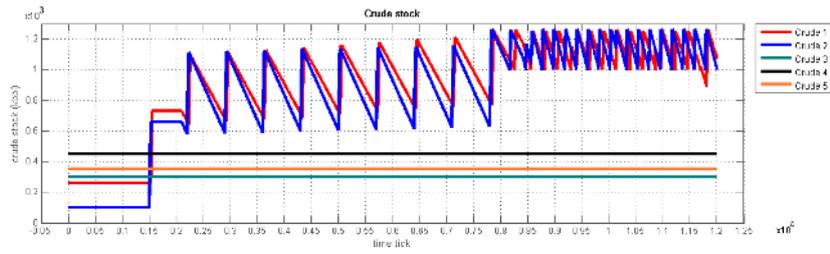
There are no numerical-method-dependent errors to be expected: In Matlab the equations are solved in a numerical way with matrix multiplications, which should always produce the same stock results as multiplications in the Repast model. Any differences in the numerical results should therefore be explained by inaccuracies in the model definition (perhaps enforced by a natural way of working in a certain paradigm).

6.3.2 Result validation

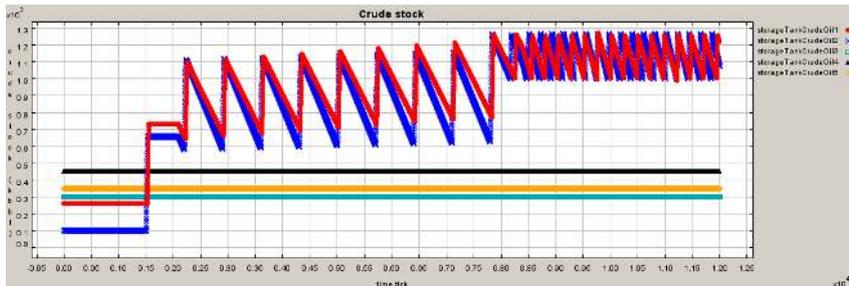
Because calculations are the same, the outcomes should be the same as well when the same input is used (this is addressed in the next step). Random numbers, however, can prevent the outcomes from being exactly the same. Stochastics are used for transport delay, demand, forecast error, etc. To prevent stochastics from influencing the outcomes, all variances can be set to 0. In this section, the following three cases are tested to evaluate result validation:

- (1) No stochastics, but fixed demands for each time step.
- (2) No stochastics as above, but halfway during the simulation the mean demands are doubled to test extreme values.
- (3) With stochastics for demand and forecast error.

For each of these three cases, the crude inventory profiles produced by both models are compared. When the two models indeed produce the same results for the base case, we can continue with more experiments.



(a) Model M



(b) Model R

Fig. 7. Results from Case 2: no stochastics with a step in demand. Time (in simulation ticks) is on the horizontal axis and the inventory level (in kbbl) is on the vertical axis.

Figure 6 shows the results of the first experiment, without any stochastics. With demand variance set to 0, during each demand cycle the same order is placed by the consumer. This requires the same crudes to be procured and the operations department plans the same production mode and throughput. Both plots show the same behaviour and the same sawtooth profiles. Even detailed dynamics are the same. For instance on day 103, the crude parcel cannot be unloaded because of lack of ullage. Furthermore, towards the end of the simulation it can be seen in both cases that there is excess crude. This is caused by a cumulative error resulting from the way the jetty behaviour is implemented in both models: the jetty has a fixed pumping rate and the smallest time unit is one tick which causes a +1% difference between the amount procured and the amount transferred to storage in each cycle (see [32] for more details). Both models show the same behaviour where pumping is paused until enough space in the crude tank becomes available. When comparing the numerical output of the stock levels at the end of the 120 days, both models produce the same number. In this scenario the models can be considered comparable.

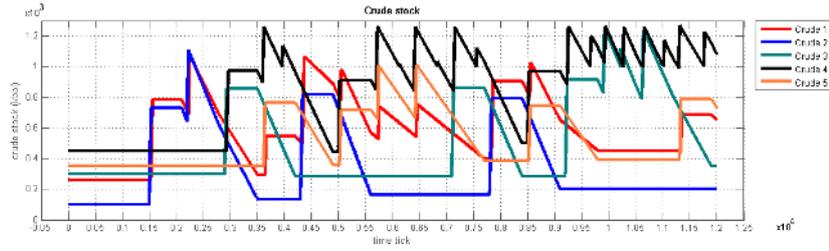
Next, in the second experiment, a step in the demand is introduced halfway during the simulation: the mean demand for all products is doubled. As could already be observed in the previous experiment, the crude stock levels are near the maximum capacity of the storage tanks, so further increasing the demand means testing extreme values for the simulation. Experimenting with such extreme values can be helpful to elucidate specific behaviour.

The results of this experiment are shown in Figure 7. The same sawtooth patterns can be observed during the start of the simulation, being exactly the same as in the previous experiment. In the second part, with extreme values, the behaviour of the models when dealing with under-capacity can be seen. Again, both models behave the same way: The two crudes are pumped alternately which creates a sawtooth pattern without any “flat” periods. The moment at which the increased demand becomes actual the effects of a higher throughput are visible. When looking at the graphs in closer detail, one can see that the crude stocks show a slightly steeper decline on day 83. This again confirms that the model results can be validated against each other.

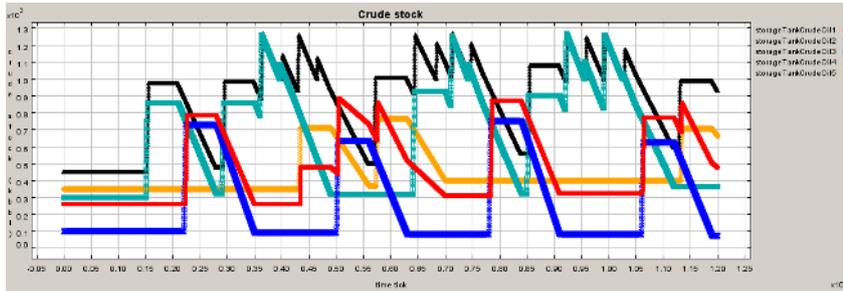
However, in this experiment small differences can be observed around day 118, so we have to conclude that the model behaviour is not exactly the same. Looking into the model definition again, a minor misalignment in the implementation of the jetty behaviour can be seen with regards to the arrival of a new ship while the previous VLCC has not been fully unloaded. This study therefore also helps the internal verification of the model by checking if the model is doing precisely what the modeller set out to build or that some (unknown) elements of the platform play a role, for example.

Finally, a new experiment is done in which demand variance is set to 25% and the forecast error to 5%. This means that during each demand cycle, different amounts of the products are ordered and that the refinery will run in various modes of operation, requiring different crudes as input for the recipe selected. Also, the amount forecasted by the sales department (and used by the procurement department to determine the amount and which crudes to buy) can differ from the actual demands, possibly causing an imbalance between the crude bought and crude used. This scenario is more realistic, because fluctuations in demand and errors in forecasting demand are part of every supply chain. A demand variance of 25% is not a realistic value though as it is higher than one would see in practice, but has been used here for modelling purposes and to amplify the effect, allowing the system to be tested in the extremes.

Figure 8 shows the results of this third experiment. At first glance, the results from the two models appear to be broadly comparable; similar saw tooth patterns can be observed between arrival of crudes when pumping happens at high speed, and the much slower release of the crudes into the distillation unit. Also the output graphs seem to fall within the same boundaries and in both cases none of the crude tanks falls empty during the simulation period. However, the random numbers drawn in the two models are different, hence a more precise match cannot be expected since different crudes are selected and different amounts are bought, resulting from different demands and forecasts of demands.



(a) Model M



(b) Model R

Fig. 8. Results from Case 3: with stochastics. Time (in simulation ticks) is on the horizontal axis and the inventory level (in kbbbl) is on the vertical axis.

6.3.3 Decision recommendations

Even though after introducing stochastics into the simulation the numerical results are not the same anymore, we can use these models and these settings to find out if the two models give the same recommendation when used for decision support.

The model output from both models (Section 6.3.2) show that the procurement policy is quite successful in that, under normal conditions without disturbances in delivery and production, crude never runs out (See experiment 3). However, the maximum capacity of the tank is reached a number of times, possibly resulting in increased demurrage costs and adding to higher storage costs. From both models we can conclude that a more *efficient* procurement policy could be found in which the volume bought is reduced and the average stock levels are lower, while still ensuring adequate crude is in stock for production.

A new procurement policy has been created in which the current stock levels are taken into account. In the standard procurement policy (described as Procurement Policy 1 in [32]), the decision of how much crude to buy is solely based on the amount that is needed to meet the forecasted demand. However, if any excess crude is still left in the storage tank after production (for example because the forecast by the sales department overestimated the actual demand), then this crude is not taken into account. The new procurement policy (referred as Procurement Policy 2 in [32]) looks at the current stock

levels in the tanks and possibly lowers the amount ordered from the supplier.

This policy has been implemented for both models. Even though, with full stochastics, the two models do not provide the exact same output, both models do recommend to use the new procurement policy, namely that the new procurement policy offers a reduction in the storage costs of raw materials. The two models each show that the average crude stocks are lower and yet the refinery never runs out of raw materials.

While the predicted impact on the profit of the refinery is not the same, both models do show that using this policy the profit of the company increases. This means that both models, when used as decision support too, give the same recommendation for this scenario. That makes the models comparable.

In conclusion, despite the fact that the numerical results are not the same when stochastics have been introduced, it is evident that the models are similar enough to continue the benchmarking of the modelling paradigms and to learn generalisable lessons.

6.4 Determination and specification of performance measures

The four performance measures described in Section 3.4 are used here:

- (1) Ease of expressing the problem
- (2) Ease of extending the models
- (3) Ease of re-use
- (4) Ease of explaining

In Section 6.6, the conclusions of the benchmarking study for each of these performance measures is given.

6.5 Description of scenarios and simulation

Performance is evaluated on a set of scenarios (well-structured experiments) that can be simulated to further compare the two models. The base case used above considers the operation of the integrated refinery supply chain over a period of 120 days. The following scenarios can be thought of as expansions to the base case:

- (1) New procurement policy (as already executed in section 6.3.3)
- (2) New production scheduling policy
- (3) Utilisation of crude storage tanks under unexpected conditions

- (4) Extra capacity CDU to deal with predicted growth in demand
- (5) Disruption in ship arrival and emergency procurement
- (6) Coping with unexpected orders

The specification of the data and parameters for each of these scenarios is reported in [32]. These six scenarios are illustrative for the value of supply chain models in decision support. The first scenario was used in Section 6.3.3 to find a more efficient policy for procurement of crudes, contributing to savings in demurrage and storage costs.

In the second scenario the decision support tool is used to try a new production scheduling policy. In the base case production throughput is based on the product with the highest crude demand, making sure that enough of this product is produced. However, this automatically means that for all other products there is an overproduction. A new production planning policy takes the amount of products already in storage into account.

Next, in the third scenario, unexpected maintenance is needed for one of the crude storage tanks, which reduces the storage capacity. This could possibly result in higher demurrage costs because there will be less space for one of the crudes. New experiments can be done in which a crude storage tank of a crude that is not used as much is converted from one type to another, so that the economically most viable crude mix can still be bought.

The fourth scenario tests the impact of a predicted growth in demand on the supply chain. To be able to better cope with this, the value of investing in extra CDU capacity is calculated so that a well-informed investment decision can be made.

The fifth scenario deals with the disruption in the arrival of a VLCC. This can have severe impact on the planned production and the delivery of end product to the consumer. An emergency procurement strategy is implemented that, in case the ship does not arrive in time, can order additional crude with a short lead time if this crude is not available in the safety stock. Experiments can demonstrate the contribution of this new policy to refinery profits.

Finally, in the sixth scenario, an unexpectedly large order by the consumer is placed (e.g. following rejection of a batch from another refinery). This is a disruption for the refinery because the product demand will be very different from the predicted amounts that the procurement and operations planning were based on. The emergency procurement strategy from the fifth scenario can be tested again in this situation to see if the refinery can cope with the high demand in the short term.

6.6 *Benchmarking conclusions*

Our results from the benchmarking study are reported next and insights in the relative advantages of the two paradigms are presented. Here the four performance indicators from Section 3.4 are revisited.

6.6.1 *Ease of expressing the problem*

Any supply chain contains two distinct types of elements:

- Production processes with complex physical and chemical phenomena (technological system)
- Decision making or business processes involving inter-entity coordination and collaboration (social system)

The behaviour of the former is best described through equations and the latter through an algorithm. The equation-based model caters well to technological aspects. The agent-based model has lesser expressive breadth for these, but offers instead a rich vocabulary for describing business processes behaviour. For example, including hold-ups of crudes in the pipes and the calculation of how much crude is transferred by the jetty were easily addressed in Model M model but were more complicated in Model R. On the other hand, the role of the 3PL and the negotiations between various shippers are easily represented in the agent-based model whereas this provides a significant challenge in Model M. Both models can, however, fully express the same problem.

6.6.2 *Ease of extending the models*

The ease of extending the model is closely linked with the ease of expressing the problem. In general, parts of the model that are easily expressed in one model, are easier to change or extend too. However, this is not always the case because it depends on *how* it was implemented. In Model M it was relatively easy to implement the behaviour of the jetty (e.g. determining the amount pumped into storage each tick) but changing it from a fixed pumping rate to a variable pumping rate is more difficult because it requires changes to the structure of the model.

Table 2: Steps required for extension of models for the scenarios described in Section 6.5.

Scenario	Changes for Model M	Changes for Model R
1	<ul style="list-style-type: none"> • Modify m-file of procurement policy: equation to calculate excess crude and equation to calculate amount of crude required considering excess crude and safety stock • Connect additional input (crude inventory) required to the “procurement_policy” function block under “Control Panel” 	<ul style="list-style-type: none"> • New algorithm to determine amount of crude needed for current cycle and compare that with current inventory • Add extra term to set of rules where amount of crude to be bought is decided
2	<ul style="list-style-type: none"> • Modify m-file of scheduling policy: equation to calculate amount of requisite demand considering product inventory and safety stock • Connect additional input (product inventory) required to the “scheduling_policy” function block under “Control Panel” 	<ul style="list-style-type: none"> • New algorithm to calculate the amount of excess products using a safety stock • The procurement algorithm should be modified to subtract this amount from the forecast demands before deciding on procurement
3	<ul style="list-style-type: none"> • Change “Crude storage capacity limit” for the two storage tanks in the “Control Panel” dialog box 	<ul style="list-style-type: none"> • The capacity of two storage tanks has to be adjusted in the knowledge base • No changes to any algorithms (or any source code) is required

Table continued on next page...

Scenario	Changes for Model M	Changes for Model R
4	<ul style="list-style-type: none"> • Add “demand_switch” constant block, “mean_demand” constant block, step block, and switch block under “Sales” • Connect the step increase and the random number generator for the forecast demand to a multiplication (product) block under “Sales”. • Change “Maximum throughput” and “Crude storage capacity limit” in the “Control Panel” dialog box 	<ul style="list-style-type: none"> • Modify algorithm of sales department to add step up in demand • A new calculation (Spreadsheet) has to be made to determine the yields of the whole refinery given the extra CDU capacity • These new values have to be entered in the instance definition in the knowledge base

Table continued on next page...

Scenario	Changes for Model M	Changes for Model R
5	<ul style="list-style-type: none"> • Add a combination of blocks to add to the transportation time simulated by the variable integer delay block under “Suppliers”, to simulate the supply disruption • Add supply disruption parameters for user input in the “Suppliers” dialog box • Write a new m-file for supply disruption emergency procurement policy • Connect the required inputs to the created “emergency_procurement_policy” function block under “Control Panel” • Add the output from the function block to the order quantity under “Procurement” • Add emergency supplier and its lead time in the “Suppliers” dialog box 	<ul style="list-style-type: none"> • An extra supplier needs to be created in the knowledge base, can use same source code as other suppliers • The shipper agent routine that checks when a ship has arrived needs extra delay factor • Shipper agent needs update of expected travel times for the new supplier • Shipper agent needs to inform refinery about the delay to trigger emergency procurement algorithm • New algorithm to determine the amount of emergency crude; Set of suppliers restricted to emergency supplier

Table continued on next page...

Scenario	Changes for Model M	Changes for Model R
6	<ul style="list-style-type: none"> • Add a combination of blocks for customer rejection to subtract from actual delivery and send low quality products back to crude tank under “Customer” • Add a combination of blocks to process low quality products into on-spec product under “CDU” • Add on-spec product to product tank balance under “Product Inventory” • Add quality disruption parameters for user input in the “Customer” dialog box • Write a new m-file for quality disruption emergency procurement policy • Connect the required inputs to the created “emergency_procurement_policy_2” function block under “Control Panel” • Add the output from the function block to the order quantity under “Procurement” • Add emergency supplier and its lead time in the “Suppliers” dialog box 	<ul style="list-style-type: none"> • Include rejection of delivered product in customer agent at given time • Update algorithm to determine actual demand (Sales department) to add product that was rejected • Add call to emergency procurement (See Case 5) when a large order is rejected

For each of the six scenarios from Section 6.5, Table 2 lists what changes are needed to be able to perform the experiment in Model M and what would have to be done in model R to do this, too. These changes are specific for Model M and Model R in that the steps are based on the implementation in Matlab/Simulink and Repast with the framework for socio-technical systems. Table 3 gives an estimate of the efforts (measured in time of one experienced

Table 3

Estimated efforts to implement the six scenarios (See Table 2) in Model M and Model R

Scenario	Model M	Model R
1	< 4 hours	< 4 hours
2	< 4 hours	< 4 hours
3	< 1 hour	< 1 hour
4	demand switch: < 1 hours change throughput < 1 hours	demand switch: < 1 hours change throughput < 4 hours
5	supply disruption: < 8 hours emergency procurement: < 4 h	supply disruption: < 1 hours emergency procurement: < 4 h
6	customer rejection: < 8 hours emergency procurement: < 4 h	customer rejection: < 8 hours emergency procurement: < 4 h

modeller) required to implement the changes from Table 2 in both Model M and Model R (while taking care that the experience in number of years of both modellers is comparable).

When looking at the estimated effort required to make these changes in either Model M or Model R we can conclude that most of these changes are easier in Model M or at least not more difficult. However, these cases were chosen with Model M in mind [32] and they do not change the structure of the model significantly. Changing a quantity is easy, but changing the representation is much more difficult.

A clear example of a difference in effort can be found in Scenario 4, where it is much more work to implement a change in the CDU capacity in Model R than it is in Model M. Equations offer an easier representation of the refinery process itself. Even though the process is modelled in both paradigms, it is easier to do this in Model M. On the other hand, for Scenario 5 it requires more effort to implement a disruption of VLCC arrival in Model M, something that is easy to do in Model R. The shipping agent already has an algorithm that checks when a ship is due to arrive as this is formalised in transport contracts so it takes little effort to make a ship arrive later than agreed upon (and, likewise, requires little effort to include penalties for late delivery, should this be required).

Model R, being a bottom-up agent-based model, has a flexible structure. The connections between the constituents (actors) is not hard-coded, unlike in Model M, hence new connections between agents can be created on the fly. One could say that more information is available in this model, because exchange or interaction between agents can be easily defined. The physical elements of

the refinery such as the CDU operation, however, are modelled explicitly in Model M and not in Model R. In summary, the two models used different *representation mechanisms* and if something is only indirectly captured in the model, it requires more effort to be changed.

6.6.3 Ease of re-use

The agent-based paradigm provides a hierarchical framework to describe the model constituents. In the framework used for Model R, a key part of the model – the ontology – was derived from earlier modelling efforts in other domains. The generic ontology makes re-use easier and also allows connections to other models, for example one of an industrial cluster incorporating other chemical industries where other agents could become consumers of the refinery. The numerical model does not *enforce* any such structure, hence reusability is in general difficult, especially between different modellers. Still, both models presented here have been used to support the development of new models in a similar, but still different, domain.

Two models of a lube additive supply chain have been created: one in Matlab/Simulink and one in Java/Repast. The main differences with the oil refinery supply chain presented in this paper are that there are multiple sites with a central sales department which assigns orders to the sites and that the process is order-based and not continuous. Furthermore, this supply chain deals with speciality products so no product inventory is kept. In the development of the Matlab model of the lube additive supply chain [43] many of the thoughts that have gone into building Model M have been re-used, but none of the actual equations or blocks have been directly transferred. The second model of this same lube additive supply chain, built in Repast [44], is more closely based on Model R. Not only does it re-use the conceptualisation, but it also shares the same ontology and through this it was possible to directly re-use source code for some of the algorithms (e.g. for procurement and the shipping agent) while adding specific scheduling algorithms that are unique to this case study.

Re-use of models has two main advantages: it saves time in the development of a new model and it helps in verification and validation of models. The first advantage is obtained when re-using the thoughts and conceptualisation and when re-using the actual source of the model. Less efforts are then required to build a new model. The second advantage is only valid when the source code is re-used: when directly using parts from a validated model (e.g. the behaviour of the shipping agent) it increases the confidence that this aspect is valid in the new model too.

6.6.4 Ease of explaining

Model M and model R can both give the same recommendations (See Section 6.3.3), but explaining how the model came to this conclusion is important too. In general one could say that equations are generally best understood by people with a background in mathematics or traditional (process systems) modelling. The agent paradigm provides a more “natural” representation that could appeal to decision makers without a mathematical modelling background. That this natural representation is only possible with agent-based models is, as we have demonstrated above, a misconception but will still be prevalent to many. Still, the explicit hierarchical structure in an agent-based model enables a natural representation for behaviours, both in terms of organisation and visualisation; this is harder with a set of equations.

The mass balance of Equation 6 could be used to explain how both Model M and model R work. Neither of the two models explicitly includes this equation, but both indirectly make this calculation. Furthermore, since Model M has a focus on *individuals* as model elements, the agent paradigm can also be used to *explain* the interactions and relationships between the actors, even if the model has not been implemented in an agent-based toolkit.

Models M and R are therefore similar in terms of ease of explaining, but Model R still has an edge for natural representation of the decision making processes and interactions between the entities in the supply chain while Model M has an edge for explaining the technical process.

7 Summary

In this paper we have demonstrated that different modelling paradigms and tools can be used to successfully create a model an oil refinery supply chain. In order to come with fair conclusions based on the comparison between them, it is important to stress that in the space of models, *equations* and *agents* are concepts of a different order. The former refers to the system description elements in the model while the latter emphasises the model elements. Thus conceptually, the “equation-based” and “agent-based” paradigms are not mutually exclusive. They are merely labels that are often convenient, but sometimes distracting. The modelling space presented in Section 2 can be used to visualise to what extent models are similar and how they are different. The three models presented in Section 5 all have different characteristics so that they are displayed in a different quadrant of the modelling space in Figure 5.

Two of these models, one created using Matlab/Simulink and one using the Repast agent platform and a framework for socio-technical systems, have been

used in a benchmarking study, following the general benchmarking steps from Section 3. By performing detailed experiments with the two models, it is demonstrated that the models are equivalent when compared using model definition, numerical results and recommended decisions. However, the modelling process itself is different for the two cases and results in different model structures and different representation mechanisms.

By analysing the efforts required to expand the models, allowing new scenarios to be tested, we identified the strengths of the two paradigms in the context of supply chain modelling. Production processes and the technological aspects are well catered by equations, while the decision making aspects can only be captured in algorithms. The complete system can, however, fully be expressed in both modelling paradigms. When it comes to extending or adjusting the models, we can say that if something is only indirectly captured in the model, it requires more effort to be changed. The physical elements of the refinery such as the CDU operation are modelled explicitly in Model M and not in Model R. On the other hand, Model R explicitly has a flexible structure, allowing new agents and connections between agents to be added in extensions to the model. In general, the efforts required to make changes in the model for a number of scenarios, ranging from operational to tactical and strategic levels, are similar.

Two models of a lube additive supply chain have been created, building upon the two models used in the benchmarking study. Many of the thoughts that have gone into building Model M have been re-used, but none of the actual equations or blocks have been directly transferred. For the agent-based model not only the conceptualisation could be re-used, but the models also shares the same ontology and through this it was possible to directly re-use source code for some of the algorithms. Finally, when it comes to explaining the model and the model results, Model R offers a natural representation of the decision making processes and interactions between the entities in the supply chain while Model M has an edge for explaining the technical process.

Acknowledgement

The authors would like to express their gratitude to the following people for contributing to the survey on the definition of agent based models: Bart de Schutter, Bri-Mathias Hodge, Catherine Chiong Meza, Chris Davis, Geert Deconinck, Igor Nikolic, Jos Vrancken, Leonid Sheremetov, Mario Verdicchio, Michiel Houwing, Paul Hines, Rudy Negenborn, Rui Duan and Weiming Shen.

This work was made possible by the Next Generation Infrastructure Foundation (<http://www.nginfra.nl/>)

References

- [1] Ian T. Cameron and G.D. Ingram. A survey of industrial process modelling across the product and process lifecycle. *Computers and Chemical Engineering*, 32(3):420–438, 24 March 2008.
- [2] E.L. Cussler and G.D. Moggridge. *Chemical Product Design*. Cambridge University Press, 2001.
- [3] K.M. Hangos and Ian T. Cameron. *Process modelling and model analysis*. Academic Press, London, 2001. ISBN 0-12-156931-4.
- [4] Ian T. Cameron, Eric S. Fraga, and I.D.L. Bogle. Process modelling goals: concepts, structure and development. In Luis Puigjaner and Antonio Espuna, editors, *European Symposium on Computer-Aided Process Engineering-15, 38th European Symposium of the Working Party on Computer Aided Process Engineering*, volume 20 of *Computer Aided Chemical Engineering*, pages 265–270. Elsevier, 2005.
- [5] Nilay Shah. Process industry supply chains: Advances and challenges. *Computers and Chemical Engineering*, 29(6):1225–1236, 15 May 2005.
- [6] Carlos A. Mendez, Jaime Cerda, Ignacio E. Grossmann, Iiro Harjunkski, and Marco Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering*, 30(6-7):913–946, 15 May 2006.
- [7] J. Forrester. Industrial dynamics: A major breakthrough for decision makers. *Harvard Business Review*, 36(4):37–66, July-August 1958.
- [8] S. Bose and J.F. Pekny. A model predictive framework for planning and scheduling problems, a case study of consumer goods supply chain. *Computers and Chemical Engineering*, 24:329–335, 2000.
- [9] E. Perea-Lopez, B.E. Ydstie, and I.E. Grossmann. Dynamic modelling and decentralised control of supply chains. *Industrial and Engineering Chemistry Research*, 40:3369–3383, 2001.
- [10] B.E. Ydstie, E. Perea-Lopez and I.E. Grossmann. A model predictive control strategy for supply chain optimisation. *Computers and Chemical Engineering*, 27:1201–1218, 2003.
- [11] C. C. Pantelides. Speedup—recent advances in process simulation. *Computers and Chemical Engineering*, 12(7):745–755, 1988.
- [12] J. Gjerdrum, N. Shah, and L.G. Papageorgiou. A combined optimisation and agent-based approach for supply chain modelling and performance assessment. *Production Planning and Control*, 12:81–88, 2000.
- [13] N. Julka, R. Srinivasan, and I. Karimi. Agent-based supply chain management – 1: Framework. *Computers and Chemical Engineering*, 26(12):1755–1769, 2002.

- [14] N. Julka, I. Karimi, and R. Srinivasan. Agent-based supply chain management – 2: a refinery application. *Computers and Chemical Engineering*, 26(12):1771–1781, 2002.
- [15] John D. Sirola, Steinar Hauan, and Arthur W. Westerberg. Toward agent-based process systems engineering: proposed framework and application to non-convex optimization. *Computers and Chemical Engineering*, 27(12):1801–1811, 2003.
- [16] B. Erik Ydstie. Distributed decision making in complex organizations: the adaptive enterprise. *Computers and Chemical Engineering*, 29(1):11–27, 2004.
- [17] Fernando D. Mele, Gonzalo Guillen, Antonio Espuna, and Luis Puigjaner. An agent-based approach for supply chain retrofitting under uncertainty. *Computers and Chemical Engineering*, 31(5–6):722–735, 2005.
- [18] H. Van Dyke Parunak, Robert Savit, and Rick L. Riolo. Agent-based modeling vs. equation-based modeling: A case study and users’ guide. In *Proceedings of the Workshop on Modelling Agent Based Systems MABS98*, Paris, France, 1998.
- [19] Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In *Proceedings of the 22nd International Conference of the System Dynamics Society*, Oxford, England, 25-29 July 2004.
- [20] Yuqing Tang, Simon Parsons, and Elizabeth Sklar. Modeling human education data: From equation-based modeling to agent-based modeling. In L. Antunes and K. Takadama, editors, *MABS 2006, LNAI 4442*, pages 41–56. Springer-Verlag Berlin Heidelberg, 2006.
- [21] Dean C. Chatfield, Jack C. Hayya, and Terry P. Harrison. A multi-formalism architecture for agent-based, order-centric supply chain simulation. *Simulation Modelling Practice and Theory*, 15:153–174, 2007.
- [22] Charles M. Macal and Michael J. North. Tutorial on agent-based modelling and simulation. In *Proceedings of the 2005 Winter Simulation Conference*, 2005.
- [23] Charles M. Macal and Michael J. North. Beer, bullwhips, and agents, October 15-16 2002. The Supply Chain Simulation Workgroup – Santa Fe Institute Business Network.
- [24] John Sterman. Testing behavioral simulation models by direct experiment. *Management Science*, 33(12):1572–1592, 1989.
- [25] M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2), 1995.
- [26] L. Monch. Simulation-based benchmarking of production control schemes for complex manufacturing systems. *Control Engineering Practice*, 17:1381–1393, 2007.

- [27] Koen Haziël van Dam, Arief Adhitya, Rajagopalan Srinivasan, and Zofia Lukszo. Benchmarking numerical and agent-based models of an oil refinery supply chain. In *proceedings of the European Symposium on Computer Aided Process Engineering ESCAPE 18*, Lyon, France, 1–4 June 2008.
- [28] Koen Haziël van Dam, Arief Adhitya, Rajagopalan Srinivasan, and Zofia Lukszo. Nuances of benchmarking agent-based and equation-based models of an oil refinery supply chain. In *Proceedings of the First International Conference on Infrastructure Systems: Building Networks for a Brighter Future*, Rotterdam, The Netherlands, 10–12 November 2008.
- [29] J.D. Sterman. *Business Dynamics. Systems Thinking and Modeling for a Complex World*. Irwin McGraw-Hill, Boston, 2000.
- [30] S. Cavalieri and M. Macchi and P. Valckenaers. Benchmarking the performance of manufacturing control systems: design principles for a web-based simulated testbed. *Journal of Intelligent Manufacturing*, 14(1):43–58, 2003.
- [31] D.J. van der Zee. Modeling decision making and control in manufacturing simulation. *International Journal of Production Economics*, 100(1):155–167, 2006.
- [32] Suresh S. Pitty, Wenkai Li, Arief Adhitya, Rajagopalan Srinivasan, and I.A. Karimi. Decision support for integrated refinery supply chains. 1. dynamic simulation. *Computers and Chemical Engineering*, 32(11):2767–2786, 2008.
- [33] Brahim Chaib-draa and Jörg Müller, editors. *Multiagent based Supply Chain Management*, volume 28 of *Studies in Computational Intelligence*. Springer, 2006.
- [34] MathWorks. Matlab, using simulink version 2, 1996.
- [35] MathWorks. Matlab 7 – programming fundamentals, 2008.
- [36] Lee Ying Koo, Arief Adhitya, Rajagopalan Srinivasan, and I.A. Karimi. Decision support for integrated refinery supply chains. 2. design and operation. *Computers and Chemical Engineering*, 32(11):2787–2800, 2008.
- [37] M.J. North, N.T. Collier, and J.R. Vos. Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation*, 16(1):1–25, 2006. ACM, New York, USA.
- [38] Koen Haziël van Dam and Zofia Lukszo. Modelling energy and transport infrastructures as a multi-agent system using a generic ontology. In *Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, The Grand Hotel, Taipei, Taiwan, 8–11 October 2006.
- [39] Ackchai Sirikijpanichkul, Koen Haziël van Dam, Luis Ferreira, and Zofia Lukszo. Optimizing the location of intermodal freight hubs: An overview of the agent based modelling approach. *J Transpn Sys Eng & IT*, 7(4):71–81, 2007.

- [40] Emile J.L. Chappin, Gerard P.J. Dijkema, Koen Haziël van Dam, and Zofia Lukszo. Modeling strategic and operational decision-making – an agent-based model of electricity producers. In *Proceedings of the 21st annual European Simulation and Modelling Conference (ESM2007)*, St. Julians, Malta, 22–24 October 2007.
- [41] Igor Nikolic, Gerard P.J. Dijkema, and Koen Haziël van Dam. Understanding and shaping the evolution of sustainable large-scale socio-technical systems towards a framework for action oriented industrial ecology. In M. Ruth and B. Davidsdottir, editors, *The Dynamics of Regions and Networks in Industrial Ecosystems*. Edward Elgar, 2008.
- [42] John H. Gennari, Mark A. Musen, Ray W. Ferguson, William E. Grosso, Monica Crubezy, Henrik Eriksson, Natalya F. Noy, and Samson W. Tu. The evolution of protege: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.
- [43] Charles Wei Kang Wong, Arief Adhitya, and Rajagopalan Srinivasan. Dynamic modeling of a global specialty chemicals supply chain. In *AICHE Annual Meeting 2008*, Philadelphia, PA, 16-21 November 2008. paper 448b.
- [44] Behzad Behdani, Zofia Lukszo, Arief Adhitya, and Rajagopalan Srinivasan. Agent based model for performance analysis of a global chemical supply chain during normal and abnormal situations. In *submitted to the European Symposium on Computer Aided Process Engineering ESCAPE-19*, Cracow, Poland, 14-17 June 2009.