INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

MODELLING ITERATION IN ENGINEERING DESIGN

David C. Wynn, Claudia M. Eckert and P. John Clarkson
Engineering Design Centre, University of Cambridge, U.K.

ABSTRACT
This paper examines design iteration and its modelling in the simulation of New Product Development (NPD) processes. A framework comprising six perspectives of iteration is proposed and it is argued that the importance of each perspective depends upon domain-specific factors. Key challenges of modelling iteration in process simulation frameworks such as the Design Structure Matrix are discussed, and we argue that no single model or framework can fully capture the iterative dynamics of an NPD process. To conclude, we propose that consideration of iteration and its representation could help identify the most appropriate modelling framework for a given process and modelling objective, thereby improving the fidelity of design process simulation models and increasing their utility.

Keywords: Design iteration, design process modelling, design process simulation

1 INTRODUCTION
Many engineering companies recognise the need to improve their development processes and thereby meet the demand for cycle time reductions. Process modelling and simulation is one way to support this, for example as a method to evaluate risks, predict outcomes and identify potential improvements.
However, New Product Development (NPD) processes are difficult to model when compared to others such as manufacturing and business workflow. This is partly due to the uncertainty surrounding design processes, which is derived from their complexity and long cycle time together with the limited overview of specialised participants [1]. Design processes are also difficult to model because the rate of process change is often significant relative to project duration, and because process outcomes depend on technical decisions that are made by examining the design *in-situ*. Most design processes also exhibit adaptive qualities, i.e., they are flexible to respond to incorporate new opportunities, requirement changes, etc. In these factors of uncertainty, complexity, novelty, *in-situ* decisions and adaptive qualities, design is thus different to repeatable processes which are known *a-priori*.
Another distinguishing characteristic of NPD is the iteration which is ubiquitous in design. Iteration is a major driver of process complexity in the development of engineered products. It is increasingly recognised as such with the move away from linear, stage-gate paradigms of the design process (see, e.g. [2] for a review) towards those such as Dynamic Product Development [3], which highlights factors such as delegation of control, the independent initiation of activity streams to reach goals, and the need for iteration in response to emerging issues and requirements [4].
Most NPD process simulations in the literature incorporate design iteration; this is usually considered necessary for an adequate model. However, despite the recognised importance of iteration in design, Safoutin argues that "*it lacks an established theoretical or operational definition, and its lay definitions are imprecise and inconsistent*" [5]. This paper contributes to the discussion of iteration and highlights the challenge of representing it in process simulation models. The ultimate aim of the research is to improve the fidelity of NPD process simulations and thereby increase their utility.

1.2 Paper overview
The paper proceeds in five sections. Six perspectives of iteration are introduced in section 2 in order to frame the discussion. Section 3 draws on industry case studies to argue that iteration is influenced by the perspectives of process participants, stage of the design process, product/process complexity and planning/management strategy. Section 4 highlights key challenges in modelling iteration. In section 5, NPD process simulation approaches are briefly reviewed and it is shown that none can capture the full complexity of iteration. Finally, it is shown that each framework is best suited to representing certain aspects of iteration and proposed that further research is necessary to explore this.

2 PERSPECTIVES OF DESIGN ITERATION

Iteration forms a recurring theme across much of design literature. However, it is often discussed as a component of a particular design process model or theory. Relatively few publications treat iteration as a research topic in its own right (see [5, 6, 7, 8] for exceptions). Although a number of frameworks have been proposed for analysing iteration, it remains particularly difficult to characterise due in part to its subjectivity. (To illustrate, a distinction might be made between 'productive' and 'unproductive' iteration to motivate the reconfiguration of design processes to reduce the latter - but a manager's understanding of productivity is likely to differ from a designer's.) Despite the evident difficulty of classification, a framework is necessary to organise the discussion in this paper. We therefore propose six non-orthogonal perspectives of iteration in design:

- **Exploration**. In modern thinking about design, it is an almost universal view that the concurrent, iterative exploration of problem and solution spaces is fundamental to the creative problem-solving process [9]. According to this solution-oriented perspective, designing involves a repeated process of solution space divergence (during synthesis) followed by convergence (during evaluation). It is the subject of many publications [2].
- **Convergence**. Many engineering problems may be viewed as the selection of parameters to meet well-defined performance objectives. Where the relationships between parameters and objectives are complex and a solution cannot be directly identified, an iterative process is used to converge upon a 'satisficing' design. Different methods and/or tools are often applied as increasing levels of detail are reached during convergence [10]. The convergence strategy is used in designer-driven processes as well as automated design and optimisation systems.
- **Refinement.** Designs which meet their primary requirements may undergo further refinement to enhance secondary characteristics, for example to improve 'elegance' or to reduce cost. Excessive refinement often occurs where it is not obvious when to stop working on a problem, for example if there are few milestones in a development schedule or if evaluation criteria are subjective. This is often the case where products have fashionable or aesthetic appeal.
- **Rework**. Tasks may require rework in response to problems that emerge as analysis is conducted, or following external influences such as requirement changes. Unnecessary rework may also be caused if the process is too complex to identify the most efficient order of work execution [11]. A configuration which eliminates this may not be possible if time constraints require work to begin with incomplete input information. Rework requires tasks to be re-attempted because their input information is updated. This is undesirable because effort is expended for no overall increase in performance or knowledge.
- **Negotiation**. Many design problems require integration of contributions from personnel who are trained in disparate disciplines and who have a limited understanding of other fields. This is common where designers do not possess a technical overview of the entire design, for example in very complex products such as aero-engines. In such cases iteration allows trade-offs between competing goals to be negotiated.
- **Repetition**. Similar tasks or steps are often performed at different points in the design cycle to apply a similar operation to different information. Repetition differs from *exploration*, *convergence*, *negotiation*, *rework* and *refinement* in that it involves re-visiting similar design activities to achieve a different goal, rather than re-visiting a goal using potentially different methods.

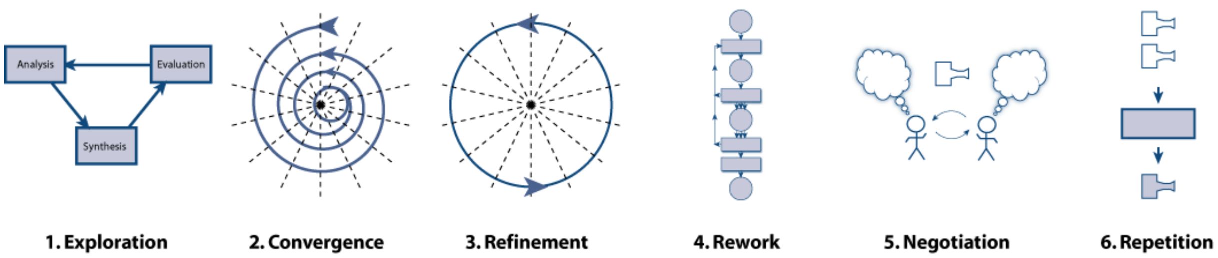

Figure 1. Six perspectives of iteration in the design process

3 INFLUENCES ON ITERATIVE BEHAVIOUR

This section discusses some influences on iterative behaviour in industry which emerged from a number of case studies. It is argued that different perspectives of iteration predominate for different participants, stages of the design process and complexity of the product/process. It is also argued that iteration is not mechanistic, but is influenced by process planning and management actions. These influences do not form an exhaustive framework; instead, the discussion is intended to highlight the complexity of iterative behaviour and its dependence upon situation-specific factors.

The studies were conducted in an aerospace company and a diesel engine manufacturer. They included observations of practice, transcribed interviews with key participants [1], development of theories and methods to support process improvement, and the construction of task-based process models based on the 'Signposting' and 'Applied Signposting' approaches ([12, 13]).

3.1 Participants' perspectives

Different process participants may have different perspectives of iteration, according to their backgrounds, responsibilities and goals. For example:

- **Component designers.** In the aerospace company, many component design processes involve specialised activities which are carried out using particular design and analysis tools. The component design process may often be viewed as the iterative application of these tools in a cycle of *convergence/refinement*. The order and manner of addressing design objectives is determined *in-situ* depending on factors such as the current state of the component, immediate design goals, time constraints and cost limitations. As the possible processes are constrained by the tools' input and output files which carry aspects of the design definition, the tools form a 'process framework' which may be modelled.

  Due to the complexity of many components and the need to compress project schedules, they are designed by teams of experts working concurrently. *Negotiation* is therefore a key aspect of component design iteration. At any time, design constraints may be altered following requirement change, either originating from the customer or to accommodate other components. Most high-performance designs comprise tightly coupled functions and cannot easily absorb such changes. As a result, any change could require *rework* of all activities – at the least to ensure their outputs are still valid. This can occur several times during a design project.
- **Project managers.** Project managers often describe processes in terms of deliverables and lead times rather than tasks and information flows. As a result projects are perceived as inter-connected concurrent workflows, organised around components, sub-systems or teams who exchange information to *negotiate* trade-offs. From this perspective, *rework* is an important influence on process behaviour. Iteration is seen as an undesirable characteristic which increases risk, lengthens cycle time and exacerbates complexity. This contrasts with the designers' perspective of iteration as a fundamental and necessary aspect of designing.
- **Specialists.** Just as many components must be designed in parallel during a project, several projects are conducted concurrently by both organisations. Design often involves a number of specialised tasks which are the responsibility of experts who address several projects concurrently. *Repetition* is thus an important component of specialists' work.

3.2 Stages in the design process

Different types of iteration are characteristic of different stages of the design process. For example, *exploration* plays a key role during early concept design as alternative solutions are proposed and evaluated. Following concept selection, work is divided into multiple concurrent streams and *convergence/refinement* dominates. Throughout this phase, *negotiation* between experts plays a key role – although because key design parameters and interfaces are frozen following concept design, negotiation is mostly confined within component/sub-system boundaries.

Activities during detailed design are often similar for different components. For example, most components require stress analysis and, depending on the organisational structure and division of responsibilities, *repetition* may therefore occur during the detail design stage. Finally, towards the end of a project engineering change may be necessitated as solutions are integrated and additional testing is conducted. Additional *negotiation* and *rework* may thus be required late in the process.

3.3 Complexity of product and process

The importance of each perspective is affected by the complexity of the product and its design process. The highly complex products in the aerospace company correspond to complex design processes which are conducted by large and geographically dispersed teams. Besides a dedicated conceptual design team, such organisations often have relatively few generalists who possess an overview of the entire product. The product is therefore designed by specialist component and sub-system teams who must interact. This requires *negotiation*-driven design which necessitates *convergence* and drives *rework* when problems occur. The individual components are often so complex that they are designed by large teams, within which experts also work concurrently and therefore engage in inner *convergence* iterations.

A key driver of iteration during the design of the less complex diesel engines arises from the need to co-ordinate projects across the organisation. A relatively small number of experts typically contribute to multiple processes, and the resulting resource limitations necessitate that many tasks begin based on early assumptions. This can lead to undesirable *rework*, a risk which is difficult to manage if it occurs at many points throughout a process. For example, routine change processes are used to generate customised versions of diesel engines. A dedicated team is responsible for most such change requests but must borrow experts from core development to address any specific problems which arise [14]. Other personnel must then either begin tasks with incomplete information and accept the risk of *rework* or downstream tasks must be delayed, a decision which is based on the perceived risk as well as the importance of timely delivery. This illustrates that complex iteration can be driven by interactions between relatively simple processes.

Although the complexity of products and processes may drive iteration this risk can be mitigated by using experienced personnel at critical points. For example, Flanagan [12] observed a transition in the engine company from a small group of experienced engineers to a more hierarchical structure and younger personnel. The more experienced engineers designed components in small teams using informal processes. They also had a clear understanding of how good a solution needed to be and could avoid unnecessary refinement. Both *convergence* and *refinement* played a greater role following the transition.

3.4 Planning and management approach

Iteration is considered during activity planning because it is expected during every design project. We observed practice ranging from explicit scheduling of a number of iterations to the utilisation of many coarsely grained, poorly integrated plans which allowed room for manoeuvre, thereby absorbing unexpected iterations [1]. Another strategy to account for uncertainty during planning is the explicit or implicit incorporation of buffers. Although useful when planning all processes, buffering is particularly important in iterative design due to the uncertainty about task duration and ordering. A practical challenge is to ensure that buffers are not repeated at many levels in the management hierarchy. This is undesirable as it may inflate schedules and can obscure the allocation of slack – which is a limited resource better reserved for absorbing unplanned delays when they occur.

In the aerospace company where *convergence* iteration played a key role, Gantt-based sequential scheduling could not account for uncertainty in task ordering. As a result, design tasks could not be scheduled in detail. This was perceived to limit the accuracy of programme monitoring and control, as progress could not be measured without a baseline for comparison. The diesel engine manufacturer has arguably less complex processes, and their 'master plans' comprised tens of thousands of tasks. However, these documents were too expensive to update following the re-planning which was necessary to respond to rework, which was inevitably caused by emerging design changes [1].

Treatments of iteration in the NPD process simulation literature often assume that uncertainty in project outcomes is derived from the complexity of interactions between activity interdependencies and individual sources of uncertainty, i.e., design processes are modelled as mechanistic interactions between activities. However, the response of a project when unplanned iteration is discovered is not mechanistic since projects are controlled by actors who reason *in-situ* about current progress, short- and long-term goals, and their predictions about future events (figure 2) [15]. For example, if prototype hardware failed during testing, management decisions would be taken to determine whether major intervention was required. Such intervention might involve obtaining more resources, negotiating relaxed milestones and/or re-planning such that delays did not accumulate but were 'reset' to a new baseline. Even without major failures, short-term goals often change on a regular basis as working plans are adapted to the dynamic project context.

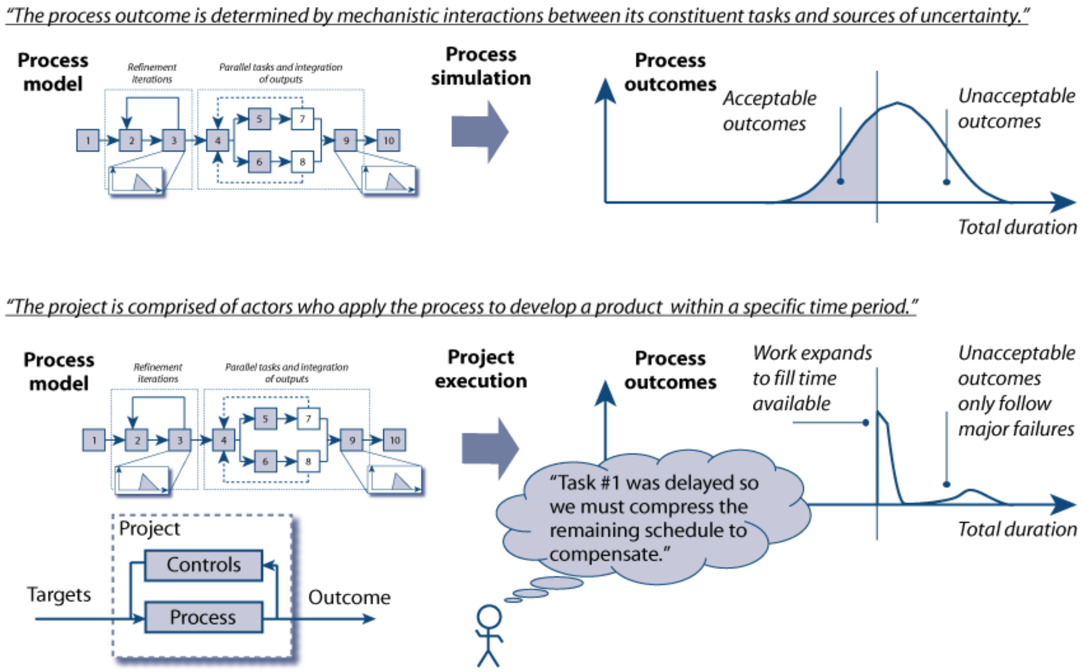

Figure 2. Iteration is not mechanistic since projects are controlled by their participants [15]

4 CHALLENGES IN MODELLING ITERATION

Previous sections have proposed six perspectives of iteration and highlighted that their importance depends on a number of situation-specific factors. The remainder of the paper focuses on the modelling of iteration. In this section, key challenges in modelling iteration are discussed.

4.2 Identifying an appropriate level of modelling detail

The term 'model' can refer to a broad range of concepts in both academic and colloquial usage. A model may be unique to an individual or shared amongst a group. It may refer to something which is tied to a particular physical medium, such as clay; a particular way of representing, such as the Gantt chart; or it may indicate an abstraction which exists only in the mind of the individual or social group. It may be formal, such as a mathematical or simulation model, or informal, such as a flowchart. In the literature there is so little agreement on the nature of models and modelling that this area has been termed the 'model muddle' [16]. However, two concepts are common to all concepts of model; a model must *represent* some target system and is in some sense *less than* that system [17]. Modellers must thus make choices about how their models abstract from the system they represent. These decisions may be of great importance, since organisational models or processes do not have a real existence and only exist in the perceptions of participants (see, e.g., [18]). From this perspective, models of a process may be its most tangible component - the way in which a process is represented can influence how people think and act beyond the intended uses of the model.

If iteration is defined as multiple attempts of an activity, its representation depends on the chosen level of description. This is because representing task-based iteration requires both perceiving a process to be composed from discrete activities and classifying two such activities undertaken at different times as similar, although they are performed upon different inputs and may involve different sub-tasks. However, the perception of similarity between tasks was found to be highly subjective during our modelling studies, depending upon individuals' roles and their understanding of the process. It also depends on the level of detail of the modelling activity and the understanding a person has of the process. Furthermore, because most tasks are ill defined and may be adequately described at varying levels of detail, the terminology or representation used to initiate a modelling exercise can influence the form of the resulting model.

To illustrate the subjectivity of representing iteration, figure 3 depicts three viewpoints of the concurrent design and manufacture of a component. From the project manager's perspective (bottom right) a convergent dialogue occurs between the design and manufacturing groups. However, the team developing the component perceives a sequence of many different tasks (left). A researcher conducting a protocol study might look closer still. From this perspective an iterative process emerges again, composed of many repetitions of a generic problem-solving process (centre).

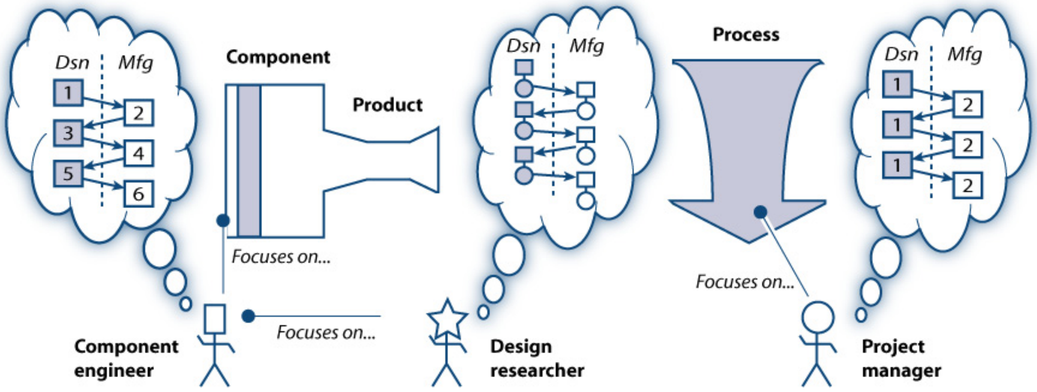

Figure 3. Iteration is subjective because it depends on dividing time into discrete activities, and classifying some of these activities as similar [15]

To summarise, whether iteration is captured in a model depends on the level of the detail and the mode of description. For example, convergence iteration that involves revisiting a number of tasks could be modelled in detail or as a single, higher-level task. Additionally, any model is likely to focus on the planned - or at least anticipated - examples of iteration. Due in part to the subjectivity of iteration, it can be difficult to identify an appropriate level of detail and to ensure this is consistent throughout a model.

4.3 Predicting the occurrence of unplanned iteration

Many activities during the design process have potential to reveal unplanned iteration. A dynamic model of process behaviour should include such possibilities. However, it is difficult to fully enumerate the potential failure points and modes of failure which drive iteration. Furthermore, even a limited list is too large and the failure modes too complex to incorporate in a task-based model. In practice, the potential failures which are considered must be selected subjectively, often based upon judgement of risk or potential impact based on recent experience. Additionally, predictions of the conditions under which unplanned iterations occur must be based on expert judgement of uncertainty; this may be biased by many factors (see, e.g., [19] for further discussion).

4.4 Predicting the consequences of unplanned iteration

Revisiting a task may invalidate assumptions and ultimately require many downstream tasks to be re-attempted [7]. For example, rig tests are usually carried out concurrently with design work in anticipation of success. If successful, the test will meet contractual obligations and provide information which may feed forward into later design work. However, the test may also indicate problems in design work which has already been completed, invalidating assumptions and thereby requiring complete or partial *rework* of many tasks. A number of factors contribute to uncertainty in identifying these tasks, including:

4.4.1 Determining which tasks must be re-attempted

The response of a project when unplanned iteration is required depends upon a number of unpredictable factors, including: the personnel working on the project and their workloads; the perceived criticality of the work; and the availability of design margin to absorb resulting changes. If insufficient margin is available, a change to one aspect of the design is likely to propagate, forcing knock-on changes to other parameters, features, components, or sub-systems to accommodate it. Such propagations commonly follow three patterns; ripples of rework which die away quickly, blossoms which are eventually brought under control, and avalanches which may require all components to be re-designed [14]. As a change to any feature may require many other tasks to be re-attempted, a key cause of uncertainty in a project's response to unplanned iteration is the need to consider the state of the product when it is discovered. In addition to this product-related uncertainty, performing unexpected rework may cause downstream tasks to be delayed [12]. The structure of information flows in a process can cause delays to be absorbed, propagated or amplified in a similar fashion to changes. This behaviour is complex and difficult to predict, even in the unlikely case where a detailed model of information flows is available [20].

4.4.2 Determining how much effort is expended on each revisited task

Each time a task is re-attempted it is likely that a different amount of effort will be expended. For example, Evans [10] describes how the effort and resources which can be applied increase with each iteration as a solution is refined. In contrast, Cho and Eppinger [21] observed that the effort required tends to *reduce* with each iteration. In general, the time expended may depend upon many factors, including the knowledge to be gained through attempting the task and the perceived likelihood of later change. Such factors cannot easily be incorporated in a model.

4.4.3 Determining how the unplanned iteration will be managed

Following discovery of iteration the outputs of any directly or indirectly dependent tasks in progress, or which have already been completed may be invalidated. The invalidated tasks cannot be fully completed until their input information has been re-generated by revisiting their invalidated predecessors. However, if a task's duration reduces upon each attempt, and if sufficient resources are available, it may be best to continue executing the task even when it is known that rework will later be required. Although continuing to work would increase the total time spent on that task, it is possible that less effort would be expended on the later attempt. This could possibly shorten the critical path of the project (e.g., [21]). Additionally, instead of repeating tasks that have been previously attempted, other tasks might be added or existing tasks amalgamated. In practice the delays associated with unexpected iteration can often be absorbed by fire-fighting and re-organising the schedule. The recent Airbus A380 project is a relatively rare example in which specific design rework was cited as the cause of major project delay. This short discussion shows that there are many possible options for managing unplanned iteration which may impact upon the project behaviour; such options cannot be parsimoniously incorporated in a simulation model which attempts to represent the dynamics of the iteration.

5 SUITABILITY OF SIMULATION MODELS TO REPRESENT ITERATION

Many approaches to support design process improvement propose methods that utilise a computable, task-based representation of a specific process to develop insights to guide improvement of that process. This section outlines the treatment of iteration in these task-based modelling frameworks and argues that no single framework can capture the full complexity of iteration highlighted above.

5.1 Approaches to NPD process simulation

A number of approaches to modelling and simulation of iterative processes may be identified in the literature. In [15] we argue they may be categorised as follows:

5.1.1 Task-based

Task-based models view the design process as composed of many tasks which are attempted – and revisited – to drive the design towards completion. Such approaches view iteration as a repetition of tasks that have already been completed, or as the execution of similar tasks in different contexts. Most task-based models are mechanistic in nature, i.e., they do not account for control mechanisms in the process. They may be classified into the following types [15]:

- **Task precedence models** such as GERT [22], Petri nets [23] and Applied Signposting [13,15] encode task ordering as pre-determined precedence relationships between design tasks. Iteration is represented using decision points and cyclic dependencies. When a decision point is 'activated' following completion of a task its outcome is determined and the simulation proceeds accordingly. The assumptions governing when iteration occurs depend upon the model; however, many are based on either simple stochastic outcomes evaluated when the decision points are reached, or on the assumption of a predetermined number of iterations. In most cases, task durations are represented as fixed values or unconditional probability density functions, sometimes configured to reduce upon subsequent attempts. The primary benefit of task network models is their ease of application, as the graphical representation reflects the logical behaviour [23]. However, this is eroded as the structure of models becomes complex, i.e., incorporating many parallel tasks, failure points and failure modes.

- **Task dependency models** including the Activity DSM [11] and IDEF0 [24] characterise iteration as a strategy to resolve information interdependencies between tasks. Several authors propose simulation algorithms based upon assumptions about the resolution strategy. The treatment of iteration depends on these algorithms. For example, Carrascosa et al. [25] discuss two types of iteration: *sequential* iteration, in which tasks which fail require revisiting one or more predecessors; and *parallel* iteration, in which concurrent tasks continuously generate rework for their coupled partners. In the former case, the process is assumed complete when no more tasks are possible; in the latter, when no tasks have any work outstanding. Cho and Eppinger extend this with the concept of *concurrent* iteration, referring to the ability to parallelise the chains of tasks which must be revisited when sequential iteration occurs, rather than to attempt each in turn [21]. A key benefit of such models is that they do not require an understanding of the baseline order for attempting tasks, as this is determined by the structure of information flows. However, they can be difficult for unfamiliar readers to interpret.
- **Adaptive task models** such as Signposting [26], MIDAS [27] and the Adaptive Test Process (ATP) [28] view design as an adaptive process which is organised around the changing state of the product. Tasks in these models are selected dynamically by considering the current state of the design at each point in the process. For example, the Signposting model assumes that tasks are selected to maximize the 'confidence' gained in that step; if several tasks would lend the same confidence, one is chosen at random. A key benefit of adaptive task models is that they do not require explicit definition of which tasks are re-attempted when iteration is required. Likewise, they also reflect that different routes may be followed on each iteration [26]. However, when a task is completed in such a model, many alternatives may be available to attempt next. The predictive utility of these models therefore depends on accurate reflection of the decisions guiding *which* of the many possible tasks are attempted at each point in the design cycle. It is often unclear whether the task selection strategies used by the adaptive models are accurate or how the resulting task sequences might be validated. These models are also difficult to visualise and hence to validate by discussion with process participants.

5.1.2 Actor-based

Design may alternatively be viewed as a process of social co-ordination, in which independent actors negotiate trade-offs arising from their different, often conflicting perspectives and goals. An actor-based perspective views iteration as a function of the continuous dialogue between process participants; such a view assumes that iteration is self-mediated and non-mechanistic. Actor-based simulations are based on modelling multiple agents and their co-ordination processes (e.g., [29]). A benefit of multi-agent models is their potential to reflect the self-organised behaviour inherent in *negotiation* and *rework*, while also modelling the detail of individual tasks. However, in comparison to the task-based models they are expensive to construct - often requiring programming – and difficult for non-experts to understand.

5.1.3 Information-based

Task-based approaches are based on an information-*processing* viewpoint in which the design process creates and refines information about the product. In contrast, an information-*based* approach treats *information about the process* as an important determinant of process behaviour, and thereby aims to capture the feedback associated with iteration (decomposition of a task into multiple iterations enables feedback by releasing preliminary information prior to its final completion). For example, the system dynamics model developed by Cooper [30] proposes that process behaviour is governed by the rework which has been generated but not yet 'discovered'. Although such models can provide a more convincing representation of project controls than task-based approaches and may be easier to construct and validate that agent-based simulations, they are typically abstract and therefore difficult for non-experts to understand. They also do not provide the task-level detail which may be sought in some modelling applications.

5.2 Suitability of simulation models to represent iteration

Each of the simulation approaches outlined above favours different aspects of iteration. However, unlike a CAD rendering of a component in which a particular feature is obscured, it can be difficult to determine whether a particular approach provides a suitable perspective of the design process. Because the modelling of iteration can strongly influence the outcome of process simulation, it is important to carefully consider the strengths and limitations of each modelling framework and ensure the selected approach is appropriate to the characteristics of a given process and modelling objective. Table 1 summarises each approach in terms of the perspectives of iteration proposed in section 2.

Table 1. Summary of key strengths and weaknesses of NPD process simulation approaches for representing design iteration

Paradigm	Approach	Strengths and weaknesses for modelling iteration
Tasks	**Task precedence, e.g.** PERT/GERT [22] Petri Net [23] ASM [13, 15, 20]	Good for convergence/refinement: • If order of attempting tasks is well-defined • Intuitive graphical notation eases communication *Poor for negotiation:* • *Cumbersome to model concurrent streams which frequently exchange information* *Poor for rework:* • *Many failure points/modes complicate flow network* • *Cannot represent revisiting different tasks on each iteration* *Poor for exploration:* • *Cannot represent in-situ task selection* *Can be difficult to manipulate/restructure diagrams*
	Task dependency, e.g. DSM [11, 21, 25] IDEF0 [24]	As above, except: Good for negotiation: • Do not require explicit model of 'flow' • Possible to model concurrent activities and: *Can be difficult to communicate meaning of models*
	Adaptive task, e.g. Signposting[26, 12] MIDAS [27] ATP [28]	Good for exploration: • Represents in-situ task selection Good for negotiation: • Do not require explicit model of 'flow' • Good for concurrent activities Good for rework • Implicitly incorporates revisiting different tasks on each iteration *Poor for convergence/refinement:* • *Difficult to represent well-defined repeating sequences or frameworks of tasks* *Difficult to visualise and therefore to communicate/validate* *Difficult to validate task selection assumptions*
Actors	**Multi-agent, e.g.** Olsen et al. [29]	Good for negotiation/rework • Represents distributed decision-making and self-organisation *Expensive to construct models, can require programming* *Difficult to understand assumptions and their consequences*
Information	**System dynamics, e.g.** Cooper [30]	• Represent feedback which governs process behaviour *Abstract approach which may not be intuitive to non-experts* *Simplifying assumptions do not represent individual tasks* *Do not offer advice regarding task-level improvements.*

5.3 Implications and opportunities for further research

This analysis highlights that certain modelling approaches predispose the modeller towards a certain perspective of iteration, and that a framework should therefore be selected by considering the characteristics of the process to be modelled as well as the purpose for modelling.

As this is a high-level predisposition it is also necessary to ensure that a model constructed within the selected framework considers the specific modelling issues outlined in section 4. These issues are sufficiently complex that it is not feasible to comprehensively represent the iterative dynamics of an NPD process in any simulation model. This would not be desirable in any case, since the practical aspects of process modelling require a relatively simple representation which can be manipulated, visualised and validated by discussion with process participants.

In conclusion, we propose that further research is necessary to explore the multiple perspectives of iteration and those factors which influence their relative importance and impact on process behaviour. We also argue that further investigation is needed regarding whether and how the challenges of modelling iteration discussed in section 4 can be overcome, how iteration impacts upon the outcome of a process simulation model, and how an NPD process simulation can be validated – or its limitations quantified and used to support conclusions drawn from analysis. It is especially important to ensure the assumptions underlying treatment of iteration are appropriate, because process model validation is usually limited by the lack of prior data for calibration and the long lead time of most NPD projects.

6 CONCLUSIONS

Although the modelling of iteration can have a critical influence on the behaviour of a design process simulation model, relatively few publications explore this in depth. This paper begins to address this by proposing six perspectives of design iteration and highlighting that, in practice, its behaviour is influenced by domain-specific factors arising from the product, the process and the perceptions of participants. It was shown that process simulation models cannot capture all possibilities for iteration in an NPD project and argued that further research in this area offers an opportunity to improve the fidelity of such models.

To summarise:

- *A framework is proposed to organise design iteration into six non-orthogonal perspectives.*

- *Iteration is difficult to describe due to uncertainty about the characterisation of design tasks, and the dependency of iterative dynamics upon complex interactions between unpredictable and domain-specific influences.*

- *Iteration is ubiquitous in design and critical in determining the dynamic behaviour of design processes. However, no simulation model or modelling framework can capture the full complexity of iteration. Further investigation into iteration and its modelling could improve the fidelity of such models and thereby increase their utility.*

REFERENCES

[1] Eckert, C. M. and Clarkson, P. J., The Reality Of Design Process Planning. *ICED'03, Stockholm, Sweden, 2003.*

[2] Wynn D. C. and Clarkson P. J., Models of Designing, in *Design Process Improvement: A Review Of Current Practice. Clarkson and Eckert (eds.) Springer 2005.*

[3] Ottosson, S., Dynamic Product Development: Findings from Participating Action Research in a Fast New Product Development Process, *Journal of Engineering Design, 7(2), pp. 151-170, 1996.*

[4] Brown, R. and Widell, R., Managing Business Processes Through Collaborative Workflow Systems. *Proceedings of the TMCE 2006, April 18-22, 2006, Ljubljana, Slovenia.*

[5] Safoutin, M. J., *A Methodology for Empirical Measurement of Iteration in Engineering Design Processes*, PhD thesis, University of Washington, USA, 2003.

[6] Costa, R. and Sobek, D. K., II, Iteration in Engineering Design: Inherent and Unavoidable or Product Choices Made? *Proceedings of DETC'03 ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Chicago, Illinois, USA,*

September 2-6, 2003.
[7] Browning, T. R., *Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development*, PhD thesis, Massachusetts Institute of Technology, USA, 1998.
[8] Costa, R., *Productive Iteration in Student Engineering Design Projects*, MSc thesis, Montana State University, USA, July 2004.
[9] Lawson, B., *How designers think*, The Architectural Press Ltd., London, 1980.
[10] Evans J.H., Basic Design Concepts. *American Society of Naval Engineers Journal, 71(4), pp. 671-678, 1959*.
[11] Eppinger, S., Whitney, D., Smith, R. and Gebala, D., A Model-Based Method for Organizing Tasks in Product Development, *Research in Engineering Design, 6, pp. 1-13, 1994.*
[12] Flanagan, T. L., *Supporting Design Planning Through Process Model Simulation,* PhD thesis, University of Cambridge, UK, 2006.
[13] Wynn D.C., Eckert C.M. and Clarkson P.J. Applied Signposting: A Modeling Framework to Support Design Process Improvement, *Proceedings of IDETC/CIE 2006 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Philadelphia, Pennsylvania, USA, September 10-13, 2006*.
[14] Jarratt, T. A., *Model-based Approach to Support the Management of Engineering Change*, PhD thesis, University of Cambridge, UK, 2004.
[15] Wynn, D. C., *Model-Based Approaches to Support Process Improvement in Complex Product Development*, PhD thesis, University of Cambridge, submitted February 2007
[16] Wartofsky, M. W., *Models: Representation and Scientific Understanding*, Boston studies in the Philosophy of Science, Vol. 129. Dordrecht: Reidel, 1979.
[17] Edmonds, B., *Syntactic Measures of Complexity*, PhD thesis, University of Manchester, UK, 1999.
[18] Checkland, P., *Systems Thinking, Systems Practice*, J. Wiley, 1981.
[19] Ayton, P. and Pascoe, E., Bias in Human Judgement Under Uncertainty, *The Knowledge Engineering Review, 1995.*
[20] Chalupnik, M. J., Wynn, D. C., Eckert, C. M. and Clarkson, P. J., Understanding Design Process Robustness: A Modelling Approach*, International Conference on Engineering Design, ICED'07, 28 - 31 August 2007, Cité des Sciences et de l'Industrie, Paris, France, 2007.*
[21] Cho, S. and Eppinger, S. D., Product Development Process Modeling Using Advanced Simulation. *Proceedings of ASME Design Engineering Technical Conferences, Pittsburgh, Pennsylvania, USA, 2001*.
[22] Pritsker, A.A.B., *GERT: Graphical Evaluation and Review Technique*, The RAND Corporation, RM-4973-NASA, April 1966.
[23] Kusiak, A. and Yang, H., Modelling the Design Process with Petri Nets, in *Concurrent Engineering: Contemporary Issues and Modern Design Tools, Parsaei HR and Sullivan WG (eds.), Chapman and Hall, 1993*.
[24] *Integration Definition for Function Modelling (IDEF0)*, Federal Information Processing Standards (FIPS) Publication 183, National Institute of Standards and Technology, Gaithersburg, MD, USA, 1993.
[25] Carrascosa, M., Eppinger, S. D. and Whitney, D. E., Using the Design Structure Matrix to Estimate Product Development Time, *Proceedings of DETC'98 1998 ASME Design Engineering Technical Conferences, Atlanta, Georgia, USA, September 13-16, 1998.*
[26] Clarkson, P. J. and Hamilton, J. R., Signposting: A Parameter-driven Task-based Model of the Design Process, *Research in Engineering Design, 12(1), pp. 18-38, 2001.*
[27] Chung M.J., Kwon P. and Pentland B.T., Making Process Visible: A Grammatical Approach to Managing Design Processes, *Journal of Mechanical Design Vol. 124, pp. 364-374, 2002.*
[28] Levardy, V, Browning, T. R., Adaptive Test Process – Designing a Project Plan that Adapts to the State of a Project. *INCOSE 2005*.
[29] Olsen, J., Cagan, J. and Kotovsky, K., Unlocking Organisational Potential: A Computational Platform for Investigating Structural Interdependence in Design, *Proceedings of IDETC/CIE 2006 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Philadelphia, Pennsylvania, USA, September 10-13, 2006*.
[30] Cooper, K. G., The Rework Cycle: How it Really Works and Reworks *PM-Network, Feb 1993.*

Contact: David C. Wynn
Cambridge University Engineering Department
Trumpington Street
Cambridge
CB2 1PZ.
United Kingdom.
Phone +44 1223 332758
Fax +44 1223 332662
Email dcw24@cam.ac.uk
URL http://www-edc.eng.cam.ac.uk/people/dcw24.html