INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

CAN WE ENGINEER BETTER PROCESS MODELS?

Udo Kannengiesser
NICTA, Australia, and School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

ABSTRACT

This paper proposes an engineering design view of process modelling that helps address the high complexity and low flexibility of many process models. In this view, processes are understood as artefacts that need to be designed, realised and adapted throughout their life cycle. The paper argues that the issues of complexity and flexibility arise as symptoms of the more fundamental problem of "delineation". This problem describes the difficulty of identifying and specifying the relationships between the various models that describe the engineering view of processes: artefact models, realisation models, and adaptation. Finally, the paper shows that the notion of design features from engineering design, represented using the function-behaviour-structure (FBS) ontology, can provide the basis for addressing the delineation problem and substantially improving process models.

Keywords: Design theory, Design methodology, Process modelling, Function-Behaviour-Structure

1 INTRODUCTION

Process modelling is a research area that deals with creating representations of processes for various purposes, including process analysis, understanding, communication, standardisation, simulation, improvement and implementation. Process models are used in a range of domains, especially in business and various engineering domains. The main application areas of these models in engineering design include the management and coordination of product design and development, and other lifecycle stages such as production, construction, maintenance and logistics. In addition, process models are tools for researchers to define design methodologies [21], and to derive product models [16].

Although a number of different, mostly graphical notations and tools for modelling processes have been developed [3], their effectiveness is often reduced due to a number of issues that remain topics of ongoing research. One of the issues is the high complexity of many process models, which significantly affects understanding of these models by human experts. A typical process diagram is shown in Figure 1, which would have to be printed on a wallpaper just to be readable. This problem is commonly perceived as a problem of model granularity, to be addressed by striking a balance between comprehensibility and precision.

Another issue is the poor flexibility of many process models. Factors such as market or strategy changes, product innovations and new regulations may require modifications of a process. Furthermore, unforeseen events in the immediate environment of the process may need to be handled flexibly, such as resource bottlenecks or effects of unexpected human or system errors. Process models that are too rigidly defined are poorly applicable in real-world contexts and are ultimately rejected by their users.

This paper explores whether an engineering design view can improve process models with respect to complexity and flexibility. The basic assumption of such a view is that processes are artefacts that need to be designed, realised and adapted throughout their life cycle, just as is the case for physical products. This view is presented in detail in Section 2. Reframing process modelling in this way allows identifying a fundamental problem, *viz.* how to delineate the different concerns and models related to process artefacts. Section 3 provides a detailed description of this problem and demonstrates how complexity and flexibility issues arise as its symptoms. Section 4 develops the beginnings of a methodology for process modelling that addresses the delineation problem, using the notion of design features and the function-behaviour-structure (FBS) ontology from engineering design. Section 5 presents related work, and Section 6 concludes the paper.

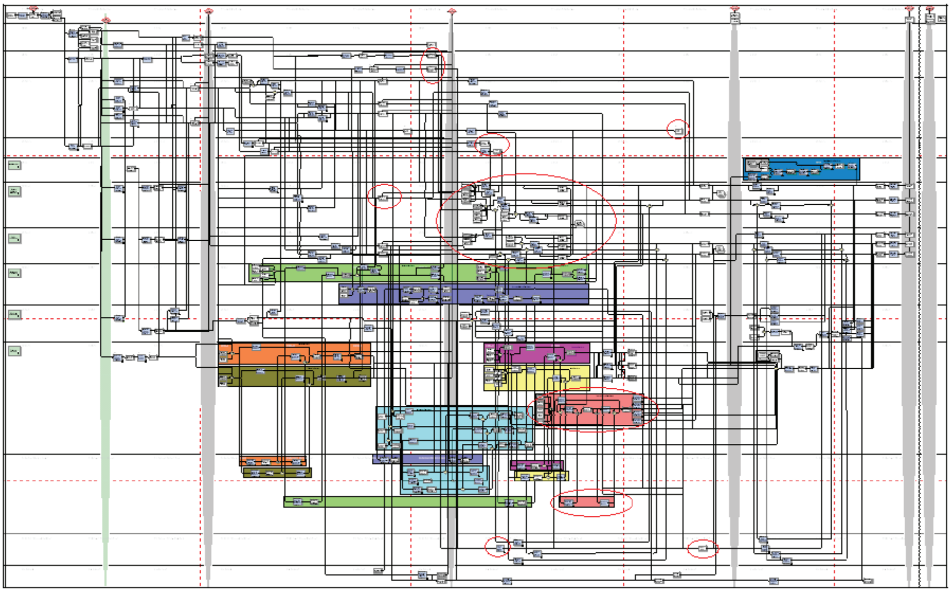

Figure 1. Example for the high complexity of many process diagrams (image from [9])

2 CONCERNS IN ENGINEERING DESIGN

There are three traditional areas of concern in most engineering design, each of which requires specific kinds of models: (1) artefacts, (2) artefact realisation, and (3) adaptation. This Section presents these concerns for both product artefacts and process artefacts.

2.1 Artefacts

Artefacts can be modelled using the FBS ontology that captures three properties: function (F), behaviour (B) and structure (S). It has been applied to various instances of artefacts, including products [8, 9] and processes [10].

- *Function* (F) is defined as an artefact's teleology ("what it is for"). Function is ascribed to behaviour by establishing a teleological connection between a human's goals and measurable effects of the artefact. An example is the function "to wake someone up" that humans generally ascribe to the behaviour of an alarm clock. The notion of function is independent of whether the artefact is a product or a process.
- *Behaviour* (B) is defined as the attributes that can be derived from an artefact's structure ("what it does"). An example of a physical product's behaviour is "weight", which can be derived directly from the product's structure properties of material and spatial dimensions. Typical behaviours of processes include measures of speed, cost, precision and accuracy.
- *Structure* (S) is defined as an artefact's components and their relationships ("what it consists of"). For physical products, structure comprises the geometry, topology and material of individual components or assemblies. For processes, structure includes three interrelated components: input, transformation and output. The transformation may be an assembly of sub-components (or sub-processes).

2.2 Artefact Realisation

Artefact realisation is a process that comprises a set of operations that transform a set of materials, components and/or sub-assemblies (input) into final artefacts (output). Both input and output of this process are embodied in the "real" world rather than a "represented" world. In most domains of engineering design, realisation is commonly known as manufacturing, assembly or construction, carried out by human workers, tools and machines. In business process domains, realisation is often called enactment and is carried out by human process workers and/or software such as workflow engines. Here, the input (or "raw materials") consists of the potential capabilities of individual human process workers and software, and the output is a coherent set of "actual" activities interpreted as the business process.

Models of artefact realisation provide very detailed accounts of the sequence of steps required for the realisation. Large parts of these models are derived from the documented structure of the artefact description. For example, the basis of assembly plans used for realising mechanical products is established mainly by product structure models (or bills of materials) generated by CAD systems. Every individual step in these plans is then associated with more detailed activities, equipment, lot sizes, time constraints etc. In addition, a number of ancillary steps may be included such as setting up tools and conveying materials within and between workshops.

In process design, models of realisation are similarly derived from the structure of the documented process artefact model. Their individual operations are consistent with the individual components of artefact structure, and their scheduling is represented by "tokens" (or states such as "started" and "completed"). Usually, the operations are at finer levels of detail than the corresponding components of the artefact. They often include ancillary steps; for example, the realisation of a business process often involves training personnel and exchanging information between different people or departments.

Generating realisation models is often viewed as an instance of designing, with the realisation process (represented in the realisation model) being the artefact. Here, the designer of the realisation process needs to consider a variety of requirements and constraints related to the realisation environment. Some of these requirements and constraints may not be explicitly specified in the architecture but are constructed from the designer's experience.

2.3 Adaptation

Adaptation refers to the activities needed for modifying the structure of an artefact, to respond to changed requirements or constraints. It is generally understood as an instance of re-designing. The structure to be adapted may not only be the structure of the artefact but also of the realisation process, since the latter can itself be viewed as an artefact. In fact, adaptation of realisation processes are quite common, to respond to unexpected events that often occur in the realisation environment. Adaptation is then concerned with recovering instances of the realisation process to avoid or mitigate possible undesired effects. This can be done by adding or removing realisation activities, re-sequencing activities, or re-allocating resources.

Adaptation, as an instance of (re-) designing, can be represented using a state space view of the FBS ontology. Here, a design state space is defined as the union of three interconnected subspaces: a function state space, a behaviour state space, and a structure state space, Figure 2. Adaptation involves modifying the variables of subspaces, the values of these variables, as well as the ranges of values. It is driven by a set of requirements and constraints, often represented as functions and behaviours, that are either explicitly specified or constructed from experience. Iterations and reformulations of the design state space occur frequently and can rarely be anticipated [9].

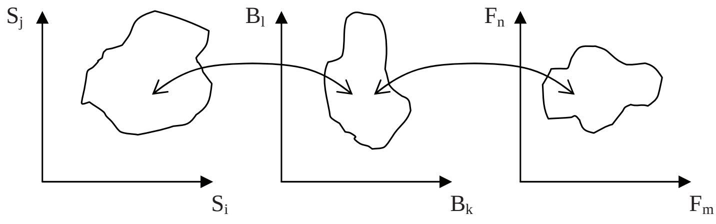

Figure 2. Function, behavior and structure state spaces and their interconnections

Take a manufacturing process of mechanical products; this process often needs to be adapted in response to variations in product demand, cost constraints, required capacity utilisation or unexpected machine breakdowns. This results in a re-designed structure of the manufacturing process, by modifying the possible kinds of manufacturing steps, their order, and the allocation of specific machines. In business process domains, the structure of realisation processes needs to be adapted to unexpected events, such as order cancellations and service interruptions, or changed constraints, such as new business rules (e.g., introduce fast-tracked handling of complaints from gold customers). The resulting changes in the realisation process may include the addition of activities (e.g., cancel shipment) and the re-allocation of resources (e.g., alternative service, and higher-rank officer in complaint department).

3 THE DELINEATION PROBLEM IN PROCESS MODELLING

While the notion of process design is widely used, the application of an engineering design view of processes has not been thoroughly explored. In particular, the three concerns presented in Section 2 have not been well understood in process modelling. We claim that this is due to what can be called the *delineation problem*: the difficulty of separating artefact models, realisation models, and adaptation, as they are all in the same domain, *viz*. the domain of processes. This Section presents details of the delineation problem and shows that some of the issues in process modelling, including complexity and rigidity, can be understood as symptoms of insufficient delineation.

As an example, we use a model of a maintenance process typical for the aviation industry. Figure 3 represents this model using BPMN (Business Process Modeling Notation; see www.bpmn.org). The process starts when the maintenance company receives a customer order. A Maintenance Planning Document (MPD) is then prepared that provides the principal basis for the next step of performing the maintenance operations. After successful completion of these operations, a Certificate of Release to Service (CRS) is sent to the customer, and, concurrently, an invoice is sent. Upon receipt of payment, the maintenance process terminates.

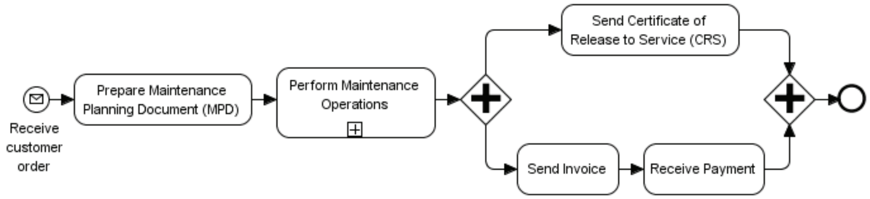

Figure 3. Example of a maintenance process

As this model is used as a vehicle for understanding the delineation problem, it only provides a very simplified view of the maintenance process. However, it is easy to see how the issues to be outlined in this Section would scale up for more detailed process models.

3.1 Artefact Models vs. Realisation Models

A fundamental assumption of process modelling has been that processes are structured sets of steps (activities or state transitions) that transform input into output. In this view, processes are often conceptualised as paths (or sets of interrelated paths based on specified conditions) along the dimension of time. Different steps within a path are activated at different times, and their activations “flow” down the specified network of paths. This view is reflected in the wide use of the term “workflow” for processes in many business contexts including engineering design. This is consistent with a view of the process model as a realisation model.

On the other hand, it is also recognised that the design of processes is driven by a set of process functions, which are often related to business or organisational goals [13, 11]. The focus here is on the front end of process use and process designs that most effectively support the desired process goals. This is consistent with a view of the process model as an artefact model.

Based on the strong similarity of the structure of artefact models and realisation models, they are often merged into a single model. This may be convenient in some cases, but can cause a number of issues. One of them is complexity, which we define here as a function of the number of process components (activities) and their relationships [4]. This problem is most prominent in artefact models, as many of the realisation details they subsume are not relevant from a design point of view. The complexity of many process models and their concomitant poor comprehensibility is a well-known issue in process modelling practice. However, this issue is often perceived as a visualisation problem [1] rather than a methodological problem. Currently, the main approach to reducing complexity is the use of hierarchical process structures that chunk some of the detailed activities into sub-processes. This is essentially an information hiding approach.

Take the "Perform Maintenance Operations" activity in Figure 3; it is represented as a "collapsed" sub-process (indicated by the "plus" sign in the lower centre) that hides all the lower-level details. Figure 4 provides an "expanded" view of this sub-process, now showing all its finer-grained activities. One of the activities in this sub-process is itself a sub-process, the details of which are shown in Figure 5. This hierarchical structure of (sub-) processes forms a "layered nesting of systems" [7], in which a (sub-) process at one level can be viewed as a realisation model of the (sub-) process at a higher level. At the same time, the same (sub-) process can be viewed as an artefact model that is realised by a (sub-) process at a lower level. Hiding all information about a particular realisation model also conceals its fundamental characteristics that distinguish it from alternative realisation models. One such alternative model for realising "Prepare Machining Fixtures" is shown in Figure 6. Here, instead of using the slower and more error-prone reverse engineering process depicted in Figure 5, this alternative uses product lifecycle management (PLM) technology to quickly access the original equipment manufacturer's (OEM's) CAD model. The fundamental differences between these realisation options and their effects for the higher-level process in terms of time and quality are significant, and should be explicitly captured in the artefact model.

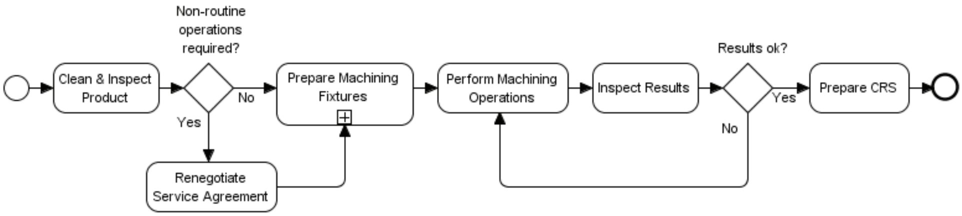

Figure 4. "Perform Maintenance Operations" sub-process

Figure 5. "Prepare Machining Fixtures" sub-process, option 1

Figure 6. "Prepare Machining Fixtures" sub-process, option 2

Using sub-processes reduces complexity only in a visual way. If no essential information is to be lost, the detailed activities can be hidden but not discarded. As a result, the overall number of components and their relationships in the process model remains unchanged.

3.2 Realisation Models vs. Adaptation

It is common modelling practice to anticipate all possible events (or exceptions) that may interfere with the process and then define ways in which the process may handle these events. A typical example is the anticipation of errors that may occur within a process step (e.g., system errors, or shortages of resources) and the definition of a new path within the process that handles these errors (e.g., by repeating the process step, or by allocating the step to a different performer). Generally, there are no reasons against this approach. However, it is often hard if not impossible to reliably predict all possible exceptions that may occur and to pre-define appropriate exception handling strategies for every exception in every situation. In addition, the "firing" of an exception depends on whether that exception is actively monitored for and with what techniques of sensing and analysis.

The explicit representation of exception handling mechanisms subsumes adaptation in the realisation model. This not only adds to complexity issues but also "fossilises" adaptation. (The "fossilisation" metaphor is borrowed from [25].) This means that the process model is made inflexible by having all possible changes of the model pre-defined. Subsequent changes are permitted only when they have been explicitly pre-defined. However, there is a need for allowing sufficient “realisation freedom” for direct stakeholders such as process performers, who are best thought of as “re-designers” of the process [34].

Take the “Perform Maintenance Operations” sub-process in Figure 4; it includes two examples of exception handling. The first example is the iteration after “Inspect Results”, which handles situations in which specific maintenance tasks could not adequately be completed. It presumes that these situations are caused by errors in performing the machining operations. While this may be a reasonable strategy in most cases, it misses the handling of those errors that are due to inaccurate machining fixtures. The second example of exception handling is the “Renegotiate Service Agreement” activity after a need for performing non-routine (i.e., unanticipated and unscheduled) operations is identified. The service agreement will always be renegotiated for this kind of exception; however, there may be different places for this activity in the process. Figure 4 depicts the particular case in which renegotiation is carried out immediately after initial inspection, and any other activity has to wait until a new service agreement is established. An alternative way of exception handling is shown in Figure 7, where the immediate response to the exception only consists of a problem log being written. The renegotiation activity is carried out later, in parallel with the “normal” flow of activities. The sub-process terminates when both flows have completed. Compared to the model in Figure 4, this mechanism can significantly increase the speed of handling the same exception. It is an instance of the “deferred fixing” exception-handling pattern proposed by [17]; its essential idea is that an exceptional situation is recognised and recorded, but dealt with later in the process.

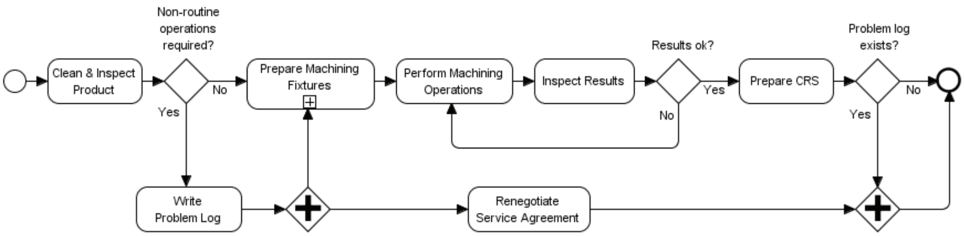

Figure 7. “Perform Maintenance Operations” sub-process, alternative exception handling

There are numerous ways in which processes can be adapted, depending on various aspects of the process situation and the growing experience of the process performer. Prescribing a “one-fits-all” exception-handling procedure will often be inadequate in real-world environments.

4 AN ENGINEERING DESIGN APPROACH TO DELINEATION

Understanding the three engineering design concerns, in the way described in this paper, can significantly improve process modelling. Table 1 summarises the key characteristics of the different concerns, providing some clues to assist process modellers delineate their models.

Table 1. Distinguishing characteristics of process artefacts, realisation and adaptation

Artefact	Realisation	Adaptation
F, B related to high-level functionality that motivates the design of the process	F, B related to feasibility, given the constraints of the realisation environment	F, B related to maintaining functionality and feasibility in dynamic environments
S related to essential, value-adding activities	S related to ancillary, supporting activities	S related to monitoring, analysing, evaluating and executing activities to prevent or handle exceptions
type-centric	instance-centric	type- or instance-centric

Dietz’ [7] work on enterprise ontology fits with some of this understanding of the delineation problem. He proposes a system of three abstraction levels, describing business processes at the highest (the “essential”) level that reflects overall business goals, at an intermediate (the “informational”) level that describes activities dealing with providing information, and at the lowest (the “documental”) level that describes activities dealing with providing documents. Process models at the essential level clearly correspond to the notion of artefact models. Process models at the informational level can then be viewed as realisation models, and process models at the documental level can be viewed as realisation models of the process at the informational level (that is now treated as an artefact model). Although Dietz’ approach is limited to business transactions, it can provide guidance for delineating artefacts and realisation in most business domains.

Being able to locate potential process components in the different models is only one condition for producing a well-delineated process (artefact) model. It is also necessary to specify the relationships between models to an extent that is necessary and sufficient for smooth and manageable interaction between the three concerns. However, most existing process modelling approaches either completely under-specify or over-specify these relationships. Can we develop an improved representation of model relationships based on an engineering view of the world?

4.1 Design Features in Engineering Design

One important outcome of research in engineering design is the notion of design features (or short, features). A feature is a description of some aspect of a product that is significant in a particular life-cycle context, such as design, manufacturing and analysis [29, 30]. Although most work in this area has focused on capturing characteristics of product structure (e.g., pockets, holes and slots), it has been proposed that features may describe any portion of the FBS representation of an artefact [2].

Design features can be viewed in two ways:

1. *Features as high-level building blocks*: Features can shorten the description of a design based on the use of standardised, high-level building blocks that are assumed to be specialised yet common knowledge [32]. This frees the engineer from having to interpret too many “unnecessary” details in the product model. It also facilitates design reasoning, as most features encapsulate semantic information represented as function or behaviour. This distinguishes these features from other building-block descriptions that encode only structure information. The notion of “feature-based modelling” is based on the idea of reducing some design problems to the configuration of features.
2. *Features as design constraints*: Features can be given to an engineer as requirements or constraints for designing or re-designing. This has the effect that the design state space is constrained in terms of variables or ranges of values for function, behaviour and structure. Designs exhibiting the required features can be viewed as “compliant” [2]. When the features have been validated through repeated, successful usage, they may become “standardised” and then reused as high-level building blocks for other designs.

The two views suggest that features have the potential to provide a generic tool for reducing complexity (through their use as high-level building blocks) and increasing flexibility (through their use as design constraints). Let us illustrate this using a simple class of features: material features. These features are often standardised and unambiguously represented using a label conforming to a convention in the materials domain. In some cases, these labels consist only of a numeral, used as an index to a more detailed description of the FBS properties of the material. In other cases, the labels reference these FBS properties more explicitly. Take the material feature labelled according to the DIN EN 10027 norm:

G-S275

This label represents cast structural steel (“G” stands for “cast”; “S” stands for “steel”) with yield strength of 275 N/mm^2. It can be interpreted as an explicit specification of the function, behaviour and structure of material:

- Function: “provide the input for casting processes”
- Behaviour: yield strength $\geq$ 275 N/mm^2
- Structure: structural type = steel

This example demonstrates the two views of features.

1. High-level building blocks: The FBS description provides high-level, semantic information rather than a low-level description of its molecular structure. Structure is referred to only through a label denoting its type ("steel"), based on its standard definition (that is assumed to be common knowledge in the materials domain) as a ferrous alloy with carbon content of less than 2.06%. Semantics is added by function and behaviour, as they provide the information relevant to product designers and product manufacturers using the material. The benefit of this representation is that it conveys rich information using a shorthand label.
2. Design constraints: The feature provides a set of constraints for the selection of material from a materials database. The feature representation supports the "principle of minimal specification" [34] for specifying only those aspects that are necessary and sufficient for the task at hand. If needed, the conventions in DIN EN 10027 allow increasing the set of constraints in a systematic, standardised way. For example, "G-S275JR" denotes the same material but with an additional constraint on behaviour, *viz*. notched impact strength of 27 Joule at 20 °C ("J" stands for 27 Joule, "R" stands for 20 °C). Note that the representation can also be used as a specification of requirements for designing new materials, e.g. as part of a systems-based approach for integrated product and materials design [23]. And the letter "G" can also be interpreted as a specification of the basic structure type of the manufacturing process (namely, "casting") required for realising the mechanical part that is annotated by this material feature. All of this shows that features can concisely and flexibly represent constraints for any area of concern (e.g., materials design or product realisation) that is different yet related to the current context of use (e.g., product design).

Another class of features includes component features. Viewed as building blocks, they represent off-the-shelf products such as nuts, bolts and bearings, or even larger components and sub-assemblies. For example, a standard engine component in a car assembly may be viewed as a high-level feature and described using a simple label rather than a detailed geometrical model. Similar to material features, component features can be represented in terms of labels that are references to more exact specifications including their detailed geometrical structure, documented in supplier catalogues. Kurtoglu et al. [14] have developed a naming convention for electro-mechanical components that is based on their functions. Component features have the same benefits as material features, as they reduce the complexity involved in modelling artefacts and increase flexibility by supporting the definition of any number of, and any type of, constraints for designing or re-designing.

4.2 Towards Feature-Based Process Modelling

Applying the idea of design features to processes can address the outlined issues in process modelling. Such an approach is promising because it would target the delineation problem directly rather than just its symptoms of complexity and rigidity. Unfortunately, most process domains do not have well-established conventions and notations for labelling the FBS properties of processes or activities. However, we can demonstrate the integration of the feature concept in process modelling using the maintenance process presented in Section 3. Figure 8 shows a feature-based model of this process, generated by annotating the activity "Perform Maintenance Operations" with semi-formal descriptions of function, behaviour and structure of the realisation process. Specifically, the Figure shows two separate feature examples.

Feature Example 1 contains a type representation of realisation structure ("maintenance operations type = MPD-confom"), whose meaning is assumed to be common knowledge among the domain experts in this example. The feature also includes information about those functions and behaviours of the realisation process that are relevant for understanding its essential characteristics. This includes the overall function to "realise 'Perform Maintenance Operations'", a quality-related function to "achieve high customer satisfaction", and the behaviour of mean time to repair ("MTTR ≤ 72h"). Feature Example 2 can be viewed as a specialisation of Feature Example 1. It augments the FBS representation provided in Feature Example 1 by adding a behaviour ("cost ≤ $50,000") and structure types related to fixture preparation ("fixture preparation type = PLM") and adaptation ("non-routine handling type = deferred fixing").

The feature-based model achieves its goals with respect to the issues of complexity and flexibility:

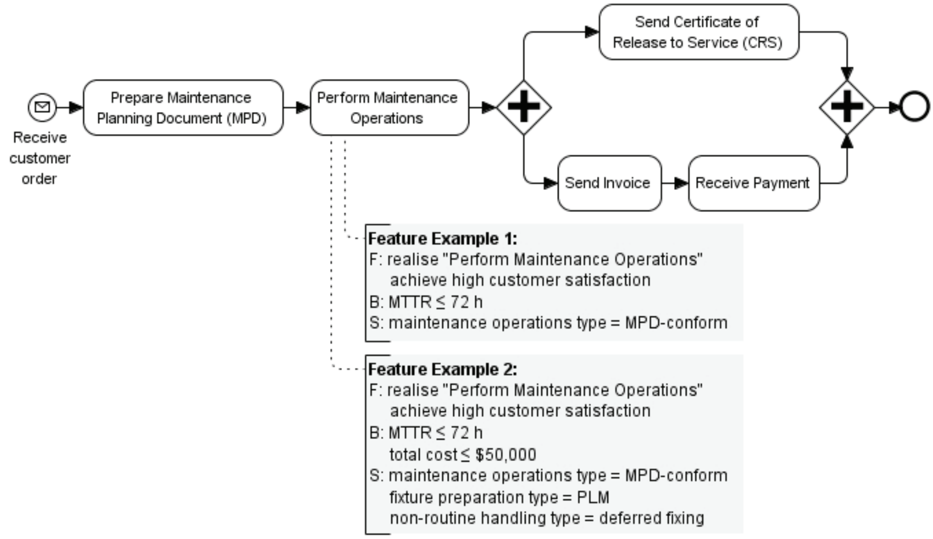

Figure 8. Feature-based model of a maintenance process

- *Complexity*: The model no longer represents "Perform Maintenance Operations" as a (collapsed) sub-process, as the delineation of the realisation model from the artefact model eliminates the need for information hiding. The information captured in the features composes high-level, meaningful building block descriptions that abstract from low-level structure details and add function and behaviour properties. While the specific information provided by the features depends on the modelling purpose and the amount of common ground between the modeller and the user of the process model, there is always a reduction of complexity. For example, while Feature Example 2 is more specific than Feature Example 1, there is no increase in complexity (in terms of the number of process components and relationships).
- *Flexibility*: Uncoupling artefact and realisation models also increases flexibility, by according stakeholders of the realisation process more space for their own variations. This "space" is the design state space, constructed both from the explicit constraints specified by features and from individual experience. The design state space can be constrained to an extent deemed necessary by the process (artefact) designer. For example, the constraints specified by Feature Example 1 would allow the use of any of the alternative realisation structures depicted in Figures 4 to 7, as long as they result in process speeds that satisfy the specified constraint on MTTR. Feature Example 2 reduces the state space of possible realisation structures to those consistent with Figures 6 and 7. Adaptation is delineated from the realisation model as the decision-making and the possible results of adapting the realisation process are no longer pre-defined as fixed exception-handling paths. However, adaptation can still be constrained, as demonstrated by Feature Example 2 that specifies a general pattern for exception handling.

5 RELATED WORK

Current research related to complexity issues focuses on the notion of process architecture frameworks that define and organise multiple views of a process, each of which represents a subset of the overall process information depending on the specific modelling objective [4]. For example, Gantt charts inform a project management view through the depiction of temporal relationships of activities and status information. The design structure matrix is a view that provides a compact schema for visualising and analysing dependencies among design tasks [33]. It allows identifying ways of improving the design process by reducing iteration and increasing concurrency. Using different views in a systematic way helps manage complexity by showing some of the attributes of process structure and hiding others. However, the structure itself must be defined in terms of all its components and their relationships, before any view can be applied. Therefore, process architecture frameworks are

unable to reduce the number of process components and their relationships, which represent the principal source of complexity.
Some approaches aim to reduce the set of components of given process structures. For example, graph reduction rules have been used for control-flow verification [27]. An approach by [24] identifies less significant activities in a process model based on measures of probability and effort. A new process model is then generated by either eliminating these activities or aggregating them with other activities. These approaches reduce model complexity, but do not extract and preserve the semantics of the pruned activities. This results in the potential loss of vital information for understanding, analysing and evaluating the process.
Other work uses standard taxonomies to incorporate domain semantics in process models. [18] investigate the use of verb classification schemes, such as the MIT Process Handbook, for uniform labelling of activities and easier comprehension of process models. Similarly, [19] describe activities in technical processes in terms of flows of energy, material and signal, based on a functional modelling taxonomy. This aims to easily identify opportunities for process automation, by mapping process activities directly onto functions of engineering products. However, these approaches lack a principled schema consistent with the FBS ontology, and thus do not provide sufficient descriptions to be used as design features.
The issue of process flexibility has been addressed by work on state space representations of processes [26]. Flexibility is understood here as the ability to move within a state space by selecting different values of the states within given ranges. Different instantiations of this concept have been proposed, including process constraints and process fragments [28], parameters for individual activities within a process [22, 31], and grammar-based representations [5]. Process modellers may then specify only the "core process", allowing for the late binding of values to variables according to individual or dynamically emerging needs. However, what is missing in these approaches is the explicit consideration of goals and requirements. Their focus has thus far been on setting up and constraining the state space of process structure but not process behaviour or process function. As a result, there is no way of specifying criteria to select appropriate process structures.
There are some approaches to supporting the search for alternative process structures. For example, signposting [6] is a technique that associates confidence values to input and output parameters of different design activities. As a consequence, the courses of action taken in a design process are not pre-defined but depend on the current state of the design. Although this approach provides useful decision support, it does not constrain the decisions in a prescriptive way. An approach by [15] integrates quality attributes, such as cost, performance and customer satisfaction, in business process models. Different processes can then be configured depending on specific stakeholder preferences. However, there is no systematic way of expressing the quality attributes as shorthand descriptions as needed for the feature concept.

6 CONCLUSION

Through the application of an engineering design view to the domain of processes, this paper has shed new light on current issues in process modelling. Specifically, it has shown that issues of complexity and flexibility can be understood as symptoms of a more fundamental problem that is specific to process artefacts, which we refer to as the delineation problem. The identification of this problem can open up a whole new range of approaches to process modelling, inspired by engineering design. These approaches can potentially have higher impact than existing ones, targeting the root cause rather than the symptoms of complexity and flexibility issues.
Real-world processes are often viewed as complex [20]. Yet, models of these processes do not have to be complex necessarily. And we can allow process flexibility more easily if we view it as an instance of designing (and re-designing) that we can control by specifying necessary and sufficient constraints. An approach from the "engineer's toolbox", *viz*. design features, can be used as a basis for a new way of process modelling in which the artefact is well delineated from other concerns. The preliminary work on feature-based process models presented in this paper has shown the potential to achieve substantial improvements in complexity and flexibility.
The applicability of this approach depends on the domain and the availability of a common ontology of the processes in the domain. Research is needed to define standards for systematic and extensible labelling of the processes that readily allows deriving their function, behaviour and structure. In addition, formal notations for representing design state spaces of processes need to be developed, so

that some of the tasks in process modelling can be supported computationally. This should include frameworks for verifying the compliance of specific process structures with structure types.

ACKNOWLEDGEMENTS

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

REFERENCES

[1] Bandara W., Indulska M., Chons S. and Sadiq S. Major issues in business process management: An expert perspective. *BPTrends*, October 2007, 1-8.

[2] Brown D.C. Features in knowledge intensive CAD: Roles & types. In J.C. Borg, P.J. Farrugia and K.P. Camilleri (eds) *Knowledge Intensive Design Technology*, 2004, pp. 53-64 (Kluwer Academic Publishers, Dordrecht).

[3] Browning T.R., Fricke E. and Negele H. Key concepts in modeling product development processes. *Systems Engineering*, 2006, 9(2), 104-128.

[4] Browning T.R. The many views of a process: Toward a process architecture framework for product development processes. *Systems Engineering*, 2009, 12(1), 69-90.

[5] Chung M.J., Kwon P. and Pentland B.T. Making process visible: A grammatical approach to managing design processes, *Journal of Mechanical Design*, 2002, 124(3), 364-374.

[6] Clarkson P.J. and Hamilton J.R. 'Signposting', a parameter-driven task-based model of the design process. *Research in Engineering Design*, 2000, 12(1), 18-38.

[7] Dietz J.L.G. *Enterprise Ontology: Theory and Methodology*, 2006 (Springer-Verlag, Berlin).

[8] Gero J.S. Design prototypes: A knowledge representation schema for design. *AI Magazine*, 1990, 11(4), 26-36.

[9] Gero J.S. and Kannengiesser U. The situated function-behaviour-structure framework. *Design Studies*, 2004, 25(4), 373-391.

[10] Gero J.S. and Kannengiesser U. A function-behavior-structure ontology of processes. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2007, 21(4), 379-391.

[11] Kavakli V. and Loucopoulos P. Goal-driven business process analysis application in electricity deregulation. *Information Systems*, 1999, 24(3), 187-207.

[12] Kreimeyer M., Eichinger M. and Lindemann U. Aligning multiple domains of design processes. In J.-C. Bocquet (ed.) *International Conference on Engineering Design '07*, 2007, Ecole Centrale Paris, France, pp. 162/1-10.

[13] Kueng P. and Kawalek P. Goal-based business process models: Creation and evaluation. *Business Process Management Journal*, 1997, 3(1), 17-38.

[14] Kurtoglu T., Campbell M.I., Bryant C.R., Stone R.B. and McAdams D.A. Deriving a component basis for computational functional synthesis. In A. Samuel and W. Lewis (eds) *International Conference on Engineering Design '05*, 2005, Melbourne, Australia.

[15] Lapouchnian A., Yu Y. and Mylopoulos J. Requirements-driven design and configuration management of business processes. In G. Alonso, P. Dadam and M. Rosemann (eds) *Business Process Management*, 2007, pp. 246-261 (Springer-Verlag, Berlin).

[16] Lee G., Eastman C.M. and Sacks R. Eliciting information for product modeling using process modeling. *Data & Knowledge Engineering*, 2007, 62(2), 292-307.

[17] Lerner B.S., Christov S., Wise A. and Osterweil L.J. Exception handling patterns for processes. In A. Garcia, C. Dony, J. Kienzle and A. Romanovsky (eds) *Proceedings of the 4th International Workshop on Exception Handling*, 2008, Atlanta, GA, pp. 55-61.

[18] Mendling J. and Recker J. Towards systematic usage of labels and icons in business process models. In T. Halpin, J. Krogstie and E. Proper (eds) *Exploring Modeling Methods for Systems Analysis and Design*, 2008, Montpellier, France, pp. 1-13.

[19] Nagel R.L., Stone R.B. and McAdams D.A. A process modeling methodology for automation of manual and time dependent processes. *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2006, Philadelphia, PA, DETC2006-99437, pp. 1-9.

[20] Osterweil L.J. What we learn from the study of ubiquitous processes. *Software Process: Improvement and Practice*, 2007, 12(5), 399-414.

[21] Pahl G. and Beitz W. *Engineering Design: A Systematic Approach*, 2007 (Springer-Verlag, Berlin).
[22] Panchal J.H., Fernandez M.G., Allen J.K., Paredis C.J.J. and Mistree F. Facilitating meta-design via separation of problem, product, and process information. *2005 ASME International Mechanical Engineering Congress and Exposition*, 2005, Orlando, FL, IMECE2005-80013, pp. 1-14.
[23] Panchal J.H., Choi H.-J., Allen J.K., McDowell D.L. and Mistree F. A systems-based approach for integrated design of materials, products and design process chains. *Journal of Computer-Aided Materials Design*, 2007, 14, 265-293.
[24] Polyvyanyy A., Smirnov S. and Weske M. Process model abstraction: A slider approach, *IEEE International Enterprise Distributed Object Computing Conference*, 2008, Munich, Germany, pp. 325-331.
[25] Ramscar M. and White R. A goal driven approach to knowledge extraction for architectural design systems. In H. Kangassalo et al. (eds) *Information Modelling and Knowledge Bases IV*, 1993, pp. 196-206 (IOS Press, Amsterdam).
[26] Regev G., Bider I. and Wegmann A. Defining business process flexibility with the help of invariants. *Software Process: Improvement and Practice*, 2007, 12(1), 65-79.
[27] Sadiq S.W. and Orlowska M.E. Applying graph reduction techniques for identifying structural conflicts in process models. In M. Jarke and A. Oberweis (eds) *Advanced Information Systems Engineering*, LNCS 1626, 1999, pp. 195-209 (Springer-Verlag, Berlin).
[28] Sadiq S.W., Orlowska M.E. and Sadiq W. Specification and validation of process constraints for flexible workflows. *Information Systems*, 2005, 30(5), 349-378.
[29] Salomons O.W., van Houten F.J.A.M. and Kals H.J.J. Review of research in feature-based design. *Journal of Manufacturing Systems*, 1993, 12(2), 113-132.
[30] Shah J.J. and Mäntylä M. *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*, 1995 (John Wiley & Sons, New York).
[31] Simidchieva B.I., Clarke L.A. and Osterweil L.J. Representing process variation with a process family. In Q. Wang, D. Pfahl and D.M. Raffo (eds) *Software Process Dynamics and Agility*, LNCS 4470, 2007, pp. 109-120 (Springer-Verlag, Berlin).
[32] Summers J.D., Maxwell D., Camp C. and Butler A.C. Features as an abstraction for designer convenience in the design of complex products. *ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2000, Baltimore, MD, DETC2000/CIE-14642, pp. 1-11.
[33] Ulrich K.T. and Eppinger S.D. *Product Design and Development*, 1995 (McGraw Hill, New York).
[34] van Aken J.E. Design science and organization development interventions: Aligning business and humanistic values. *Journal of Applied Behavioral Science*, 2007, 43(1), 67-88.

Contact: Udo Kannengiesser
NICTA
Managing Complexity Theme
Locked Bag 9013
Alexandria NSW 1435
Australia
Tel: +61 2 9376 2202
Fax: +61 2 9376 2203
E-mail: udo.kannengiesser@nicta.com.au
URL: http://www.nicta.com.au/people/kannengiesseru

Udo Kannengiesser is a member of the software engineering research team at National ICT Australia (NICTA). He obtained his PhD at the University of Sydney (Australia), and a degree in mechanical engineering at the University of Karlsruhe (Germany). His research focuses on models of designing products and processes, the FBS ontology, situatedness, and agent-based models of interoperability.