Journal of Engineering Design
Vol. 22, No. 3, March 2011, 145–164

Process and event modelling for conceptual design

Robert L. Nagel[a]*, Ryan Hutcheson[b], Daniel A. McAdams[b] and Robert Stone[a]

[a] *School of Mechanical, Industrial and Manufacturing Engineering, Oregon State University, 204 Rogers Hall, Corvallis, OR 97331, USA;* [b] *Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA*

(*Received 18 December 2008; final version received 5 June 2009*)

Abstractions perform a fundamental role during product design, freeing a problem from reality into a representation more readily represented with engineering principles. Functional modelling provides such a representation for product design where customer needs are translated into a representation of elementary operations defining what a product must do to achieve a desired goal. With solely the generation of a functional model, the designer, however, runs the risk of failing to explicitly capture expected interactions and operations of the product as a whole. To that end, this paper presents a process modelling methodology consisting of model levels for the representation of product-related events and configurations based on current functional modelling techniques. Being based on functional modelling allows process modelling to integrate with functional modelling during conceptual design activities. The levels for the process model then collectively define customer needs related to how a product will be used, environments where a product will operate and changes that a product must undergo to meet customer expectations. To demonstrate the generation of event and configuration models, a common household product is investigated; this is followed by a case study discussion where process modelling is applied during the design of two ground robots.

Keywords: abstraction; process; event; configuration and functional modelling

1. Introduction

Abstractions perform a fundamental role in the problem solving process. Without appropriate abstractions, many complex problems could not be solved. An abstraction, by definition, allows a problem to be extracted from its physical reality in such a way as to provide a means for problem solvers to solve those problems, which prior to abstraction, exist as nothing more than an idea (Webster's Encyclopedic Unabridged Dictionary of the English Language, 2001, The New Oxford American Dictionary, 2005). Abstractions take various forms in all branches of engineering. For instance, free-body diagrams generated in mechanics are used to solve equilibrium problems by extracting a particle from its surroundings and applying known force loads. Electrical schematics provide abstractions of circuits with symbols to represent components such that a circuit may be readily understood, analysed and constructed. Control engineers generate block diagrams linking dynamical attributes of a system to gain understanding of overall system dynamics. Computer-aided design (CAD) provides numerous potential abstractions for engineers and designers to

*Corresponding author. Email: rlnagel@designengineeringlab.org

ISSN 0954-4828 print/ISSN 1466-1837 online

DOI: 10.1080/09544820903099575
http://www.informaworld.com

convey information to managers, customers and marketing. The abstractions developed in solid modelling utilising CAD technology also provides a means to package complex systems, analyse stresses and isolate failures. Broadly, each of these abstractions may be considered a model, where a model, as defined in *Webster's encyclopedic unabridged dictionary*, is 'a simplified representation of a system or phenomenon, as in the science or economics, with any hypotheses required to describe the system or explain the phenomenon, often mathematically' (Webster's Encyclopedic Unabridged Dictionary of the English Language, 2001).

In engineering design, abstractions tend to be models developed to answer specific questions during the development and evaluation of solutions. Many design texts discuss the importance of applying this type of abstraction to expand a designer's view of a design problem, enhance creativity, clearly define design goals and decouple designs into distinct sub-problems (Cross 2000, Otto and Wood 2001, Ulrich and Eppinger 2004, Voland 2004, Pahl *et al.* 2007). To that end, functional modelling is considered a key part of the engineering design process (Miles 1961, Hundal 1990, Dieter 1991, Cutherell 1996, Otto and Wood 2001, Ullman 2002, Ulrich and Eppinger 2004, Pahl *et al.* 2007). Functional models, when applied during preliminary design, provide flexible models for problem abstraction, which can help answer numerous design questions with a focus 'on *what* has to be achieved by a new concept or redesign and not *how* it is to be achieved' (Otto and Wood 2001). An abstraction, such as a functional model, based on *what* a product must do instead of *how* it will be done provides the benefits of an explicit relationship to customer needs, comprehensive understanding of the design problem, enhanced creativity, innovative concept generation and systematic organisation of both design problems and the design team (Otto and Wood 2001, Ullman 2002).

Functional models tend, however, to concentrate on a single decomposition of a physical artifact where models focus *inside* of the product on how the product will operate. But what happens when the customer needs require designers to focus *outside* of the workings of the product on its actions? Perhaps the customer needs lead to a product capable of performing unique operations for different applications? Perhaps the product is expected to interact explicitly with its environment? Or, perhaps the customer desires a product designed to account for different configurations such as how it will be stored, operated, cleaned or maintained? To that end, a process modelling technique is developed in this paper to address customer needs related to how the final product will be used by the potential customer. Process models, generated during conceptual design, are meant to assist designers with developing an understanding of what outcomes the customer is trying to reach, what operations the customer expects the product to perform and what design configurations are required.

The process modelling technique is based on current functional modelling techniques and is developed with the functional-process modelling integration in mind. As process models detailing customer–product interactions increase in fidelity, they begin to capture elements related to product functionality. While this might create some redundancy in model elements, it also facilitates a natural integration between function and process where the differentiation between function and process is maintained by the perspective of how various elements are captured. The process modelling methodology presented extends the prior work of the authors found in Hutcheson *et al.* (2006) where mission critical events are modelled and in Nagel *et al.* (2006) where human-centric operations are modelled. This process modelling methodology presented herein extends this prior work to develop a consistent process and event modelling approach based on traditional functional modelling methods for conceptual design applications. The organisation of this paper is as follows: First, a review of related approaches for modelling and analysing processes is provided. This review is followed by definitions related to the process and event modelling nomenclature used throughout the remainder of this paper. A simple example is utilised throughout the following methodology section to demonstrate the generation of a process model. The relationship between functional and process models is then explored. A case study example based on two robotic ground vehicles follows, and the paper ends with a discussion and concluding remarks.

2. Background

As abstractions generated during the design process grow to model configuration-based product operations along with user and environmental interactions, the line between traditional design techniques and process-based project management techniques begins to blur. Within the realm of project planning, authors (Goslin 1967, Randolph and Posner 1988, Hunt 1996, Williams 1996, Darnton and Darnton 1997, Spinner 1997, Rosenau and Githens 2005, Westland 2006) advocate looking at an overall task to ensure completion of the desired deliverables; this is analogous to product design where the goal is to develop solutions to assist customers in meeting their need to more effectively complete their task at hand (LeBoeuf 1987, Hipple 1988, Otto and Wood 2001, Christensen and Raynor 2003, Ulrich and Eppinger 2004, Ulwick 2005). In project planning, a five-step work force planning model guides the creation of a successful project plan; the model includes: (1) identifying project direction, (2) analysing the workforce, (3) determining a plan of action, (4) implementing the plan and (5) continual monitoring (US Office of Personnel Management 2001). Traditionally, the product design process also consists of five general steps, which can, again, be considered analogous to project planning. The product design process begins by guiding the designer to assess the customer's needs (i.e. identify the direction), which is followed by formulating the problem (i.e. analysing what must be done). Abstraction and synthesis follow; it is during this phase where a designer begins to develop various plans of action for how to meet the customer's needs. The various plans of action are analysed during the fourth phase, implemented during the final phase and iterated continuously through the process (Voland 2004). It is during these initial phases of the design process – often broadly termed conceptual design – where process and event modelling is meant to assist designers with a product's design.

In the realm of project planning, programme evaluation and review technique (PERT) (Malcolm *et al.* 1959), critical-path method (CPM) (Kelley and Walker 1959), workflow planning (Ju 2001) and workflow activity models (WAMO) (Eder and Liebhard 1995) provide modelling approaches for capturing project milestones, flow of project resources and basic scheduling information. PERT directs the process of generating models consisting of boxes to represent tasks and flows to represent resources. Boxes are labelled with respective completion times and are formatted in a linear progression such that parallel and sequential tasks are intrinsically represented (Malcolm *et al.* 1959). CPM captures activities, their durations and dependencies to mathematically schedule a workflow for project activities such that critical actions are fulfilled (Kelley and Walker 1959). Workflow planning, as proposed by Ju, defines each process as a single workflow comprised of one or more sub-processes (Ju 2001). Managers are free to coordinate and make decisions on the sub-processes within each workflow. WAMO, like the aforementioned techniques, is based on the general idea that any business process can be broken down into smaller working units or workflows, which can again be broken down into smaller sub-processes (Eder and Liebhard 1995). A key difference, however, is that Eder and Liebhard propose visualising each workflow as an activity tree such that a hierarchy is created where activities contain activities.

Process flow diagrams, like the aforementioned project planning techniques, have been developed to model workflows. Process flow diagrams, instead, focus on the workflow of the design team. A block diagram details the teams' actions during the design process and can include critical review points, programme approval goals and other design milestones (Ulrich and Eppinger 2004). A similar approach, proposed by Andersson *et al.* (1998), more rigorously models the design process providing a framework for the application of sensitivity analysis within the design process modelling structure. Model elements capture tasks and their characteristics as well as design reviews and their probability for success.

Design structure matrices (DSM) are also used to capture and model design processes by providing forward and backward dependencies to represent task and resource relationships of a design process (Ulrich and Eppinger 2004). Optimisation strategies may then be used for evaluation of process execution strategies (Cho and Eppinger 2001). A difficulty, however, with the application of a DSM for design process modelling is the elicitation of thorough and accurate process information; to overcome this difficulty, Wallace *et al*. propose an open-marketplace approach to collect process information and auto-propagate a DSM (Wallace *et al*. 1999). The auto-propagated DSM can subsequently be optimised following DSM-based approaches. A DSM has also been utilised as a tool to tie the design process to the evolution of a design (Fagerström and Nilsson 2003) with an integrated modelling structure to capture evolving product functionality and solution strategies, and the product development process is modelled following integrated definition method #0 (IDEF0).

The IDEF consists of a suite of function-based modelling tools, which provide a framework for developing function and process models. Traditionally, IDEF0 is the method utilised to generate system function models (NIST 1993), while IDEF3 is utilised for process representations (Mayer *et al*. 1995). IDEF models utilise highly structured methodologies for the representation of functionality, actions and organisation relating to a system. In a similar vein to the IDEF approach are unified modelling language (UML) and systems modelling language (SysML). Both UML and SysML provide specifications for a number of different model types to represent complex system requirements and behaviour as activities and sequential events, as states for specific system use cases (Friedenthal *et al*. 2008, OMG – Object Management Group 2009). Of the model types, the intent of use case diagrams most closely resembles process and event modelling. Use case diagrams, which are the same in both UML and SysML, represent how a system is used to accomplish a goal when influenced by an external entity – termed an actor (Friedenthal *et al*. 2008). Use case diagrams are used as a part of a use case system decomposition where functionality is derived from analysis of how the product will be used; the use case diagrams are often derived from the system requirements. Use case models, like the majority of the models defined within UML and SysML, are based upon the behaviour and functionality of the system as a whole, and these models' formats, in conjunction with the largely information-based flows, make UML and SysML well suited for modelling information heavy systems. While these approaches may be considered function-based, the model formats are complicated and include different implementation structures for each model type, which tends to require custom computational frameworks for their application. This, along with their focus on information-based flows, makes both methods well suited for information management applications; however, limits their usefulness in early product design applications.

Functional approaches typically are limited to a single product configuration and do not capture process information. To address this concern, a number of functional decomposition approaches have been proposed which include a representation for product behaviour (Gero 1990, Umeda *et al*. 1990, Tor *et al*. 2002, Zhang *et al*. 2002, Goel *et al*. 2009). In these approaches, behaviour is a principle-centric representation with a focus on how the final product will or should act in its final implementation; function is then derived from or is directly related to the behaviour of the product. In process and event modelling, however, flows required for the process models, like flows required for functional models, are derived from the customer needs and represent a way to capture the customer interactions, product configurations, product actions and environmental circumstances, which would otherwise not be modelled following traditional functional decomposition approaches. To more rigorously model these human–product interactions, stand-alone methodologies have been proposed. Activity diagrams allow for the mapping of the interactions between user and product with boxes representing tasks. Tasks are connected sequentially through flow lines (Otto and Wood 2001). Process trees may also be applied to generate a cradle to grave diagram for the possible applications of a product (Otto and Wood 2001). Neither process trees nor activity models, however, have a rigidly defined structure, which limits their interoperability with functional modelling approaches.

Each of these aforementioned approaches, while being applicable during the product design process, fails to effectively model the customer's expected usage of the product in such a way as to be integrated with existing function-based abstractions that are currently being used during the product design process. Instead, the methodologies focus on design team interactions, milestones during the design process, design evolution or management of design information. While important to the design process as a whole, these methods are more applicable for scheduling the design team, evaluating progress, managing product and design information and planning product redesigns. An approach based on current functional abstractions and process planning is required to more fully capture expected customer actions, product interactions and operational environments.

3. Process and event modelling definitions

Formally, process models may be defined as a set of events with identified inputs and outputs occurring across time where each event focuses on an individual product configuration. This is similar to the definition of a functional model where functions define the transformations of identified input flows into desired output flows. In process and event modelling, these transformations occur as a part of product configurations, and consequently, each of the product's configurations, as modelled in a process, can be further decomposed and modelled as a functional model. It is important to note that function in this paper is used to represent the transformation of input flows into desired output flows. This definition differs from the term functional requirement (FR). FRs represent a more general definition encompassing the product as whole and generally do not define or address the transformations of flows through the final product (Suh 1990). FRs tend to be used in matrix-based design representations rather than the model-based discussed in this paper. To help clarify these nuances of nomenclature, the following definitions related to process and functional modelling are provided:

- Operation: A mechanical or natural progression or procedure, which is a part of a productive activity whereby work is expelled to affect a desired outcome (Webster's Encyclopedic Unabridged Dictionary of the English Language, 2001).
- Process modelling: The overall approach to modelling a series of customer-driven, product-based operations related through input and output flows, the product being designed and time.
- Functional modelling: The overall approach to modelling *what* a product must do in terms of elementary operations such that the product may achieve an overall goal or purpose (Stone *et al*. 2000).
- Flow: A material, energy or signal, which interacts with the product; flows are expressed as nouns (Stone *et al*. 2000).
- Function: A description of an operation, expressed as the active verb, performed by an artifact, as a part of a larger product, to transform an input flow to a desired output flow (Stone *et al*. 2000). Functions are tied together via material, energy and signal flows.
- Configuration: A specific discrete instance of the overall function of the product occurring as a part of an event. The configuration of a product is modelled functionally.
- Event: A set of configurations of a product, which may relate to the environments where the product is used, changes to the operability of a product, specific applications of a product or sequencing of operations during the usage of a product.
- Process: The sum of defined events that occur with respect to the product as a whole and aim to meet a particular goal. Processes are tied together via the product, material, energy and signal flows.
- Time line: A representation of the temporal relationship between events and configurations in a process model.

- Black box (process) model: The high-level process model defined by a single overall event representing the task to be accomplished.
- Event model: A more detailed process model consisting of multiple events that collectively define the customer's operations with the product.
- Configuration model: A detailed model of the individual operations and changes occurring to the product as a whole and involved in completing a particular event.
- Functional model: A structured arrangement of functional elements tied together via material, energy and signal flows describing the operation (or functionality) of an artifact or collection of artifacts that collectively comprise the product.

Process and event modelling adds a layer of breadth to the depth provided through the use of formal functional modelling techniques by enabling increased fidelity of modelling abstractions generated during product design. Functional modelling provides depth through hierarchical models of single individual configurations of a product. Process modelling provides breadth to functional abstractions by capturing customer–product interactions through the unique events that a product is expected to operate within. In this sense, a functional model is a sub-set of a process model where process models capture the discrete changes a product must undergo through both spatial and temporal domains to deliver a desired final outcome. Thus, process models and functional models share a hierarchical relationship that allows for a more accurate representation of the functional operation of a product.

4. The process modelling methodology

A process model provides a technique to abstract the operations performed by a customer on a product during the conceptual design phase. These abstractions provide a means to detail changes to product configurations through a series of events occurring over time. Configurations directed towards a common end goal sum to an event, and events sum to a black box to create the hierarchical relationship shown in Figure 1. Overall the sum of the black box, events and configurations abstract the operations expected of the final customer such that he or she may achieve the desired outcome defined by a customer needs set.

The decomposition of a process into a process model, shown diagrammatically in Figure 2, follows a methodology similar to that of the generation of a functional model, where first, the

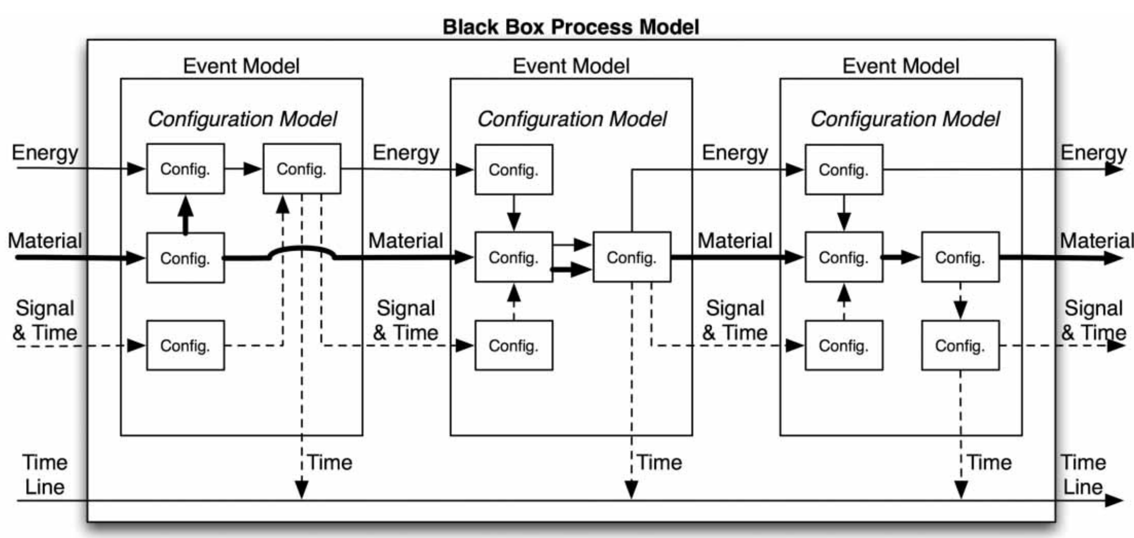

Figure 1. Hierarchy of process, event and configuration models.

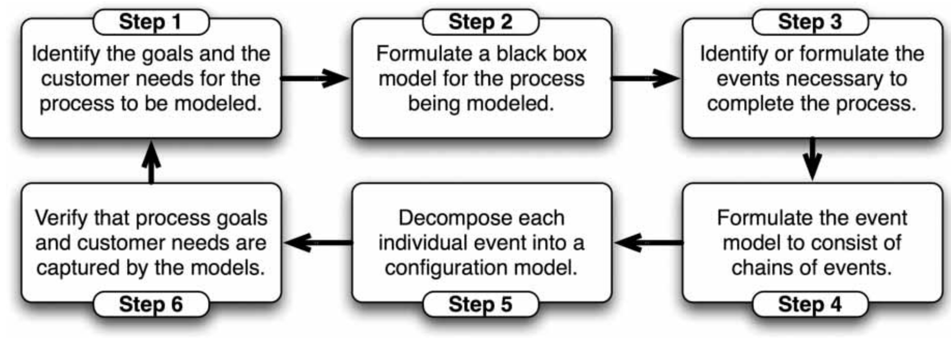

Figure 2. Process modelling methodology.

customer needs and process requirements are identified. Then from these needs, the high-level process is modelled as a single black box. Individual events are identified and then are formulated into chains of events to decompose the high-level black box model. The event chains collectively define the operations required of a product such that the outcome desired by the customer may be reached. Each event model can be further decomposed into a configuration model focusing on specific changes to the product. Process models contain at least three successive layers of detail: black box, events and configurations. The following sub-sections detail the generation of each of these model layers.

4.1. *Generating a black box model for the process*

At the highest level of detail, a black box model defines the overall process that the customer expects to accomplish with the product being designed. The black box model is derived from the requirements for the process including the customers' overall process requirements, needs and tasks, the process goals and the desired outcomes. This black box model, shown in Figure 3, takes the form of a single event. Input and output flows identify all elements required to complete the process, and along with start and stop times, follow the traditional functional modelling convention where materials are bold arrows, energies are single weight arrows and signals (including time) are dashed arrows. These three types of flows are drawn entering and exiting the black box model of the process. The following first and second steps relate to the generation of a black box model:

Step 1: Identify the overall process associated with the product being designed. Identify the requirements – including the customer needs, tasks, goals and outcomes – for the process and product as well as the available material, energy and signal flows.

Step 2: Formulate a black box model for the process being modelled based on the requirements identified during Step 1. This model defines the overall process for the usage of the product being designed along with its associated energy, material and signal inputs and outputs.

To demonstrate the development of a process model, consider the design of a gourmet-style coffee maker. The first step to generating a black box model for the process of brewing coffee with the

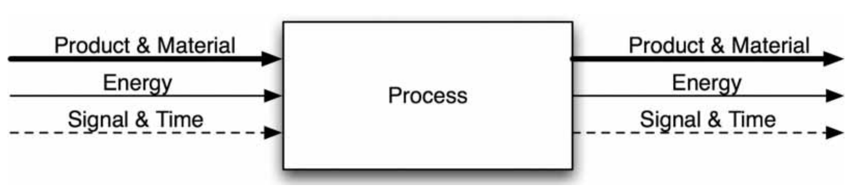

Figure 3. Example black box model for a process.

coffee maker is to understand the customer's expected outcomes and objectives. Of course, the customer's expected outcome is to brew a cup of coffee; however, to focus the modelling, let us narrow the overall objective to the brewing of 12 cups of coffee from whole beans. A number of alternatives could have been selected at this point such as to use coffee pods, ground or powdered coffee, or even liquid coffee concentrate, but since the coffee maker is suppose to be a gourmet product and to stick with the gourmet theme, whole beans have been selected as the source for the coffee's flavour. To accommodate the whole beans, the coffee maker will need to combine a coffee bean grinder with a coffee maker; through this paper, this combined product will be called a grind and brew coffee maker. To brew the coffee requires that the coffee maker be first, setup, and second, run.

The information collected during the first step is now utilised to compile a black box model for the process of brewing coffee. The black box, shown in Figure 4, models the overall desired outcome – brew coffee – and the associated energy, material and signal flow requirements. Flow conservation shows that the flows of water, beans and electricity are converted to the flows of coffee, grounds, steam, thermal energy and acoustic energy through the brewing process.

An important distinction between process and functional models, demonstrated in Figure 4, is the inclusion of the product – coffee maker – in the process model. With functional models, the product is not included as a distinct flow; the focus of such models is the set of transformations occurring *inside* the product and the product cannot act upon itself. With process models, however, the focus of the abstraction is *outside* of the product, and it is important to include the product as a flow. The inclusion of the product as a flow allows the designer to explicitly model customer and environmental interactions with the product as well as product configurations. This is a primary distinction between traditional functional modelling techniques and the proposed process modelling technique.

4.2. *Generating an event model for the process*

The event model decomposes the overall process, as represented by the black box, into a chain of events that must occur in order for the customer to achieve their desired outcome. The third and fourth steps specify this decomposition of the black box model into events:

Step 3: Identify or formulate the events necessary to complete the current or proposed process using the requirements identified in Step 1. For each event, identify input and output flows, start and stop times and required product configurations such that the product can be transformed to meet the customer's specific requirements.

Step 4: Formulate the event model to consist of chains of events that must be completed systematically to achieve the desired goal or outcome. The event sequence should begin with the initial product action and should be followed by all other discrete events including each of

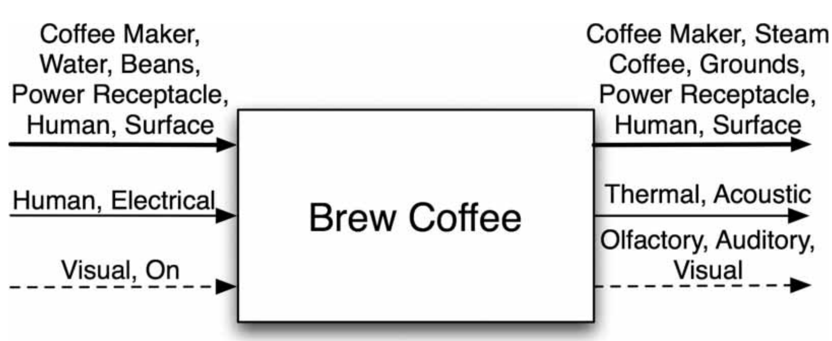

Figure 4. Black box model for the process of brewing coffee with the grind and brew coffee maker.

the actions, environments or situations where the product will be used over time. The product should be included as a flow through each event.

Following Step 3 for the coffee making process, the events identified include: a basic setup including stabilising and powering the appliance, adding coffee beans, adding water and brewing. The flows include: a coffee maker, surface, electricity, coffee beans, water and a human. Specific event start and stop times are probably not important; however, it is important to recognise that the events occur sequentially.

Once the individual events and their required flows are identified, Step 4 is followed to generate chains where each element in the chain resembles a single black box containing input and output flows of materials, energies and signals as demonstrated by Figure 5. The first event of the model starts with the initial action or operation of the product, and each progressive element in the event model identifies new operations that must occur as time progresses. Temporal information is captured with either signal flows or a time line. A time line is drawn parallel to the process model with the initiation time at the left and the completion time at the right. Times marked along the time line are tied to each event with a dashed signal flow. Figure 5, however, shows time in both formats for illustrative purposes.

In an event model, there are interevent flows that are required for more than one event and intraevent flows that are required for only one particular event. Interevent flows are typically drawn entering and exiting the vertical (left and right) sides of an event box, where intraevent flows are typically drawn entering and exiting the horizontal (top and bottom) sides of an event box. Flows can also skip an event or feedback to an earlier event depending on the customer needs and requirements. Figure 5 shows a material flow feedback between the third and first events. Step 4 prompts the decomposition of a black box model into an event model.

Generation of event models for the grind and brew coffee maker leads to the decomposition of the black box model into four specific events – setup coffee maker, load coffee beans, load water and operate coffee maker – shown in Figure 6. The input flows for the coffee maker are human material, human energy, electrical material (the power receptacle) and electrical energy. Each of these flows are required through all of the events, thus they are not listed as exiting the product until after the completion of the final event. Beans, water, and the on signal are not needed until their appropriate event, so each are imported at their respective event vertically. To represent time, the process model is drawn parallel to a time line connected to each process event via a signal flow. Time increments up following the junction of each signal flow with the time line.

4.3. *Generating configuration models for each event*

Configuration models, developed for each event in the process, are the most detailed level of a process model. Configuration chains should be made for all of the input flows (materials, energies and signals) as well as for the product. Each chain should capture all of the individual

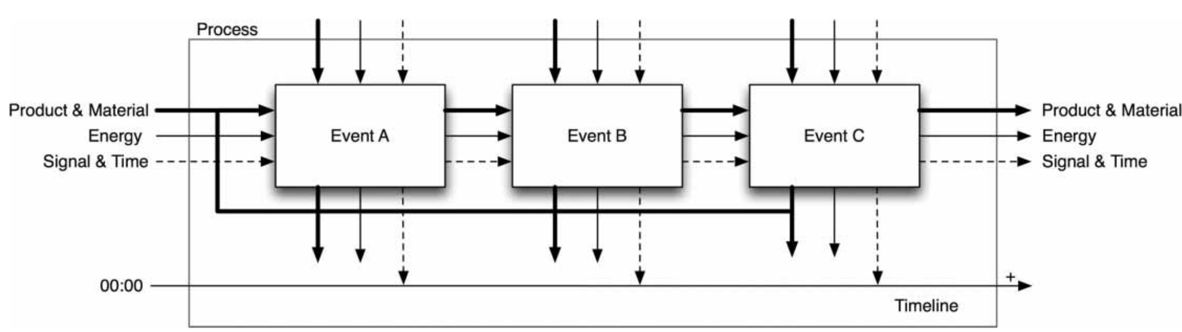

Figure 5. Example event model for a process.

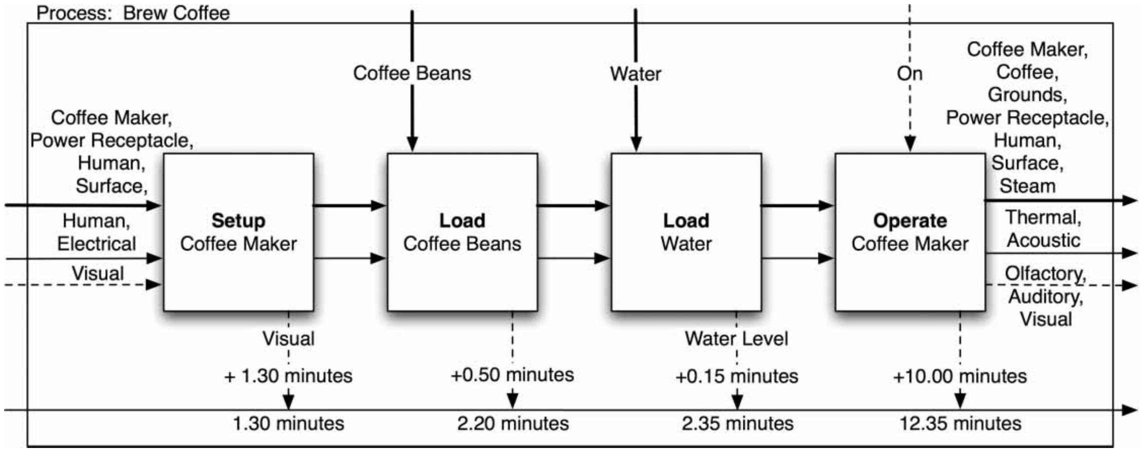

Figure 6. Event model for the process of brewing coffee with the grind and brew coffee maker.

changes to each flow and product configurations that must be achieved in order to arrive at the desired output(s). Once each of the chains is produced, they are aggregated to create a complete configuration model as shown in Figure 7. The generation of configuration models is guided by Step 5:

> Step 5: Decompose each individual event in the process model into a configuration model detailing the discrete changes to the product and any associated functional interactions with other flows in the event. As necessary, time should be represented as either a time flow or as a time line. If a time line is used to represent the flow of time, signal flows may be used to connect each configuration to the time line.

For the grind and brew coffee maker, consider the event of loading coffee beans as an example for the generation of configuration models. The configuration model for the loading of coffee beans, shown in Figure 8, provides an abstraction for the coffee maker twice being adjusted via a configuration change guided by the motion of the operator's hand. These configuration changes represent the various bean chambers being opened for coffee bean storage. Coffee beans are mixed with the coffee maker, and then the grind chamber lid is secured back into position. A time line drawn along the event model indicates that it should take approximately 15 seconds to change or

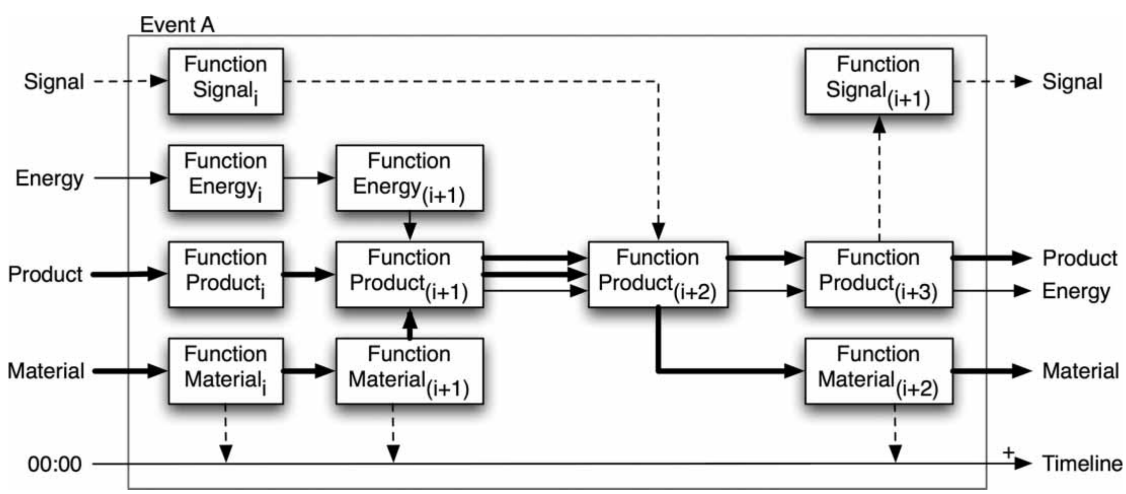

Figure 7. Example configuration model for an event.

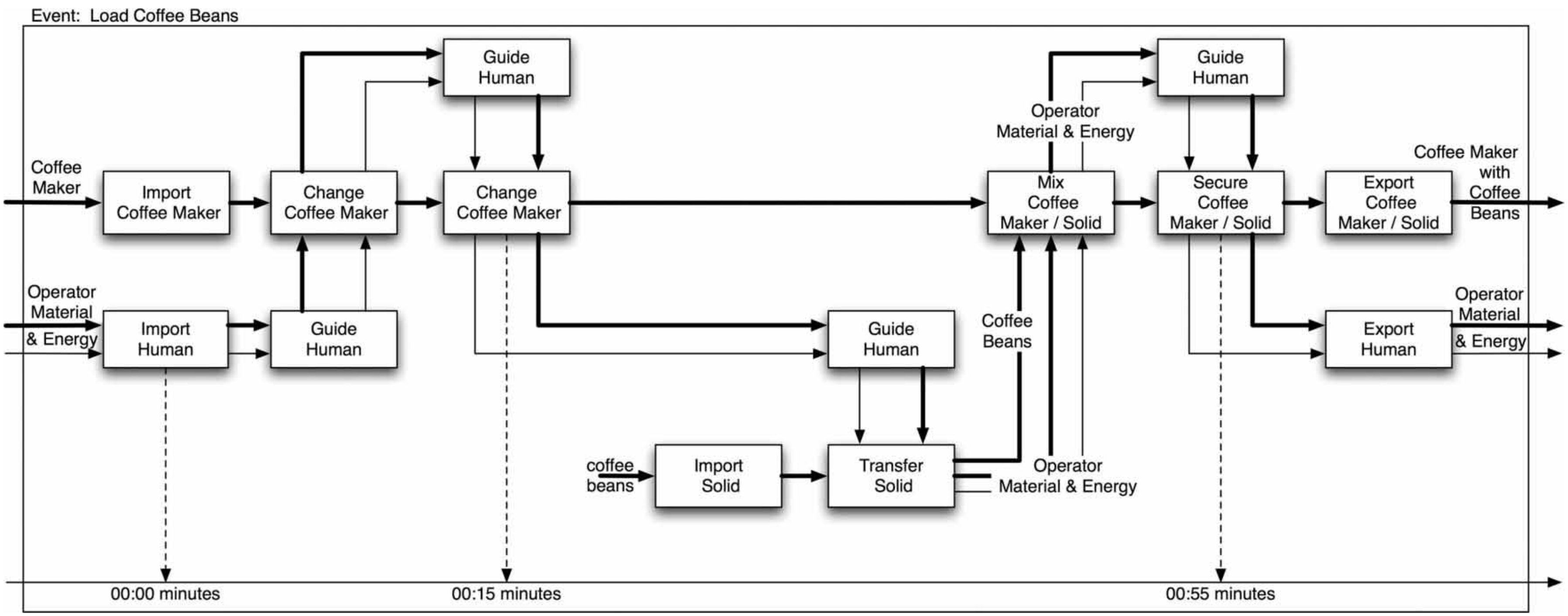

Figure 8. Configuration model for the event of loading beans into the grind and brew coffee maker.

to ready the coffee maker to accept coffee beans, and approximately 55 seconds to complete the filling of the coffee bean chamber.

Each configuration model for the remaining events in the process of brewing coffee with the grind and brew coffee maker are generated similarly to the event described above for loading coffee beans. Once all of the models have been made detailing the process, it is important to verify that the initial customer needs prompting the generation of the models are addressed by following Step 6:

> Step 6: Verify that the process models generated address all of the customer's requirements identified in Step 1, abstract all expected operations and model the achievement of the customer's expected outcomes stemming from the application of the product being designed.

For instance, the customer's goal – stated in Step 1 – is to brew coffee in the grind and brew coffee maker starting from whole beans, thus in Figure 8, whole coffee beans are imported into the event, mixed with the coffee maker and secured into the grinder compartment.

4.4. *Process and functional representations*

Process models, once developed, are an extension of traditional functional modelling as defined in Otto and Wood (2001) and Pahl *et al*. (2007) and as such should be considered a part of the functional analysis process. Process models may be generated either prior to or following the generation of functional models to provide insight into the customer's goals and expectations for a product. This insight should lead to iteration of the functional models. Process models also may provide a framework to model product configurations where functional models are created for each configuration of the product modelled through the process. The result of combined functional and process modelling is a complete description of the customer's expected functionality of the product through each action the customer expects the product to operate.

For the grind and brew coffee maker, a sub-functional model, developed traditionally following Otto and Wood (2001) and Pahl *et al*. (2007), is a conglomeration of all physical product configurations through each event in the process. A sub-functional model of the coffee maker, shown in Figure 9, first imports the required coffee beans, water and electricity. Electrical energy is converted into mechanical energy to grind the coffee beans, which once ground, are stored waiting to have their flavour extracted. The extract is mixed with heated water to create coffee. The resultant coffee is stored with a supply of warming thermal energy for future consumption.

With configuration models, process models begin to blur the boundaries between process and functional modelling. Depending on how the product's boundaries have been defined, functionality may be shared between sub-functional and configuration models. This occurs because of the hierarchical relationship between process and functional modelling shown in Figure 10 and is most evident with the modelling of flows from the environment that interact with the product yet are expected for successful operation. Environmental flows may be imported into both the configuration and sub-functional models of the system and have functionality in both models dealing with the flow. It is important to remember, however, that the overall perspective between a process and function is different, and this difference should be reflected in the modelling of the flows shared between configuration and sub-functional models.

For instance, both the sub-functional and process models represent the coffee beans being coupled with the system, but the perspective taken by the models is different. The functional representation includes the importation of coffee beans to represent the beans going into the coffee bean chamber to be subsequently ground and harvested. In the configuration model, however, the importation of coffee beans represents the customer bringing the coffee beans to the coffee maker, which has to be changed to accept the coffee beans into a grinder chamber. While this difference

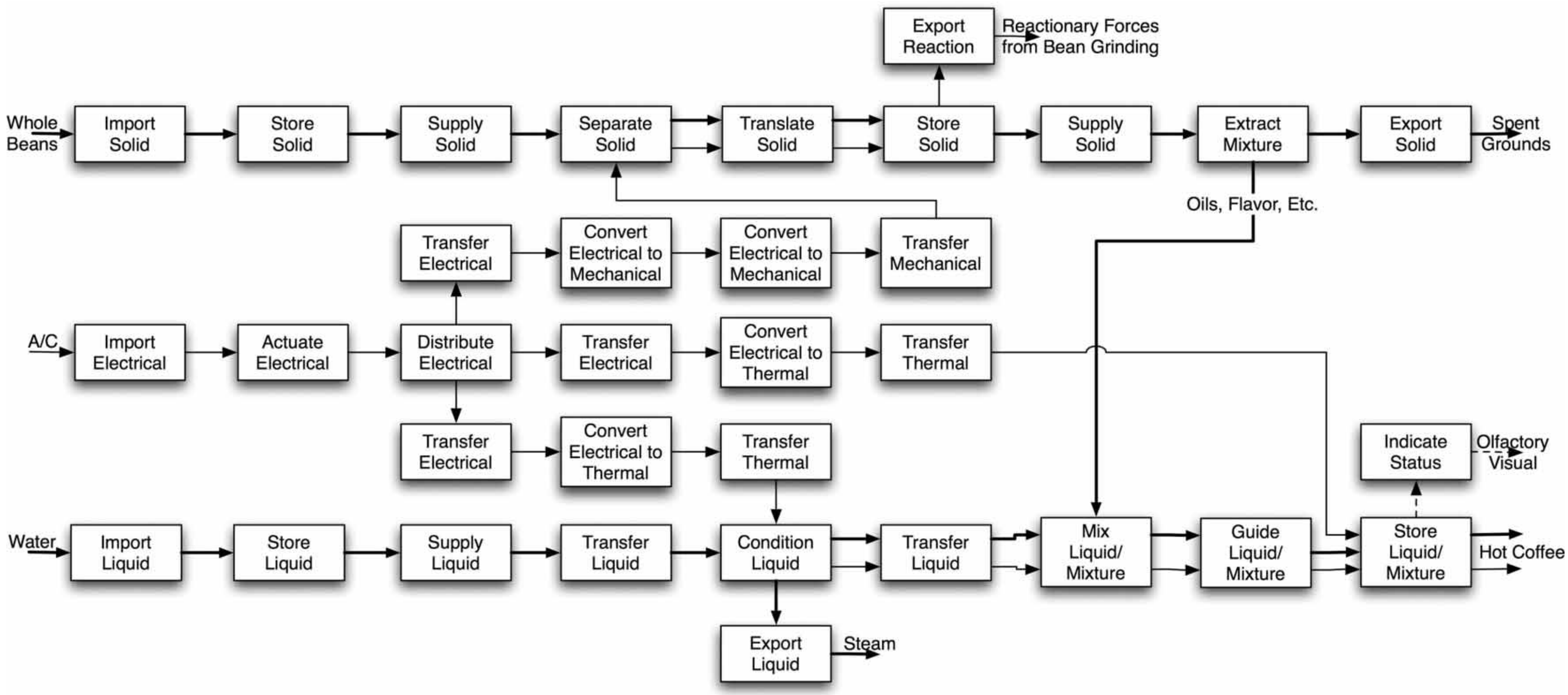

Figure 9. Functional model for the grind and brew coffee maker.

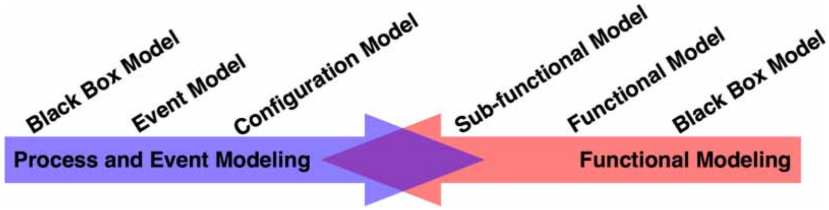

Figure 10. The relationship between functional and process models.

is indeed small, it is important to note that the coffee maker, in all of the process models, is represented wholly as a flow – the product flow – and all interactions occur with this product flow. Including the product flow allows the designer to freely model changes and interactions specific to the product as a whole based on the customer needs, and differentiates the perspective between process and functional models where process models exist as a layer above the functional models. Thus, the functional models provide depth with a focus on the inside of the product, and the process models provide breadth with a focus on the product as a whole.

5. Case studies – the design of intelligent ground vehicles

To assess the usefulness and applicability of the process modelling approach presented in this research, the methodology is applied during the conceptual design of two robotic ground vehicles as a part of the course work in a graduate-level design course at Missouri University of Science and Technology. Each robotic vehicle is designed by separate design teams – Team A and Team B – for different customers with different requirements. Team A designs their vehicle to compete in the navigation challenge at the Intelligent Ground Vehicle Competition (IGVC). Team B designs their vehicle for the United States Army to perform remote explosive ordinance disposal (EOD).

The design process outlined by Otto and Wood (2001) is followed by both teams through the conceptual design of the robotic ground vehicles with the exception that process modelling is used in conjunction with functional modelling to more fully explore the requirements of each team's design problem. The design process begins with both teams gathering and understanding their customer's needs. Needs are translated to design objectives and utilised to generate functional models. Following the generation of functional models, the teams are tasked with generating process models following the approach outlined in this paper. The process models are used to gain a deeper understanding of the environments where their robots will be operating, the setups required and storage and maintenance requirements. Once the models are generated, the groups are asked to iterate their functional models to reflect any new information learned from their process models.

5.1. *Team A – the IGVC robot designers*

The information captured by the two teams in their process models well reflects the information available to them during the design process. Team A – the IGVC robot designers – is able to read well-defined competition rules and to interview students who helped with prior IGVC robots. From this information, Team A develops a complete decomposition of process models starting with a black box model abstracting the task of navigating the obstacle course. The black box model is decomposed into a string of three events: calibrate robot, setup robot and operate robot. Team A's event model, shown as Figure 11, includes an iteration loop for calibrating the robot to the course environment, a setup event to load the payload and orient the robot on the obstacle course, and an operate event where the robot is autonomously navigating the obstacle course.

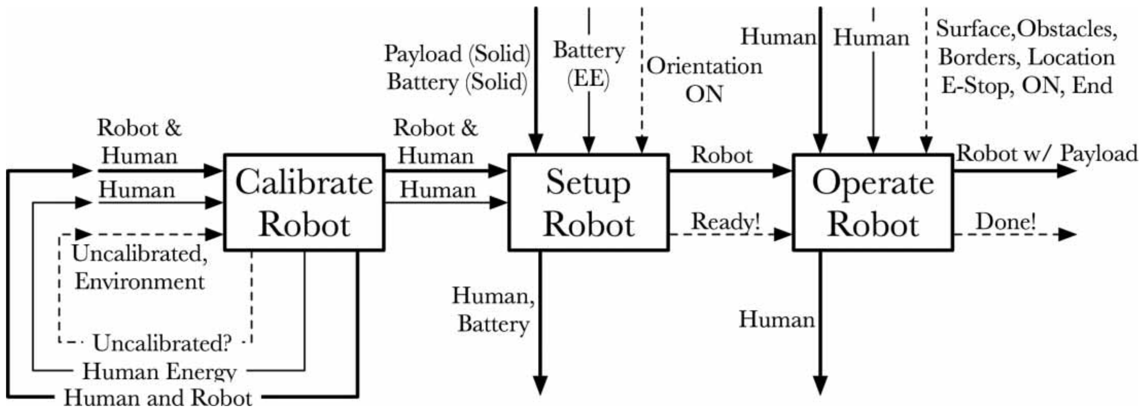

Figure 11. Event model generated by Team A for the process of competing in the navigation challenge at the Intelligent Ground Vehicle Competition (IGVC).

Team A is able to further decompose each event into a configuration model by considering how they want their design to accomplish each of these required tasks. For instance, the configuration model for the setup event, shown as Figure 12, models the human replacing drained batteries with a charged set, placing the payload into the robot and moving the robot onto the obstacle course for the competition.

Team A subsequently updates their functional model to represent what is learned during the process analysis. From the configuration model of the calibrate robot event, Team A decides that the functional model should explicitly include the ability to initiate a calibration procedure. From the configuration model for the setup of the robot, the team realises that the robot must be able to switch between an autonomous travel mode and a manual travel mode for placement on the obstacle course, and that the functional model should explicitly model the payload compartment to insure simple loading of the payload into the robot. And, finally, based on the configuration model of the operate robot event, Team A decides to better clarify the product and environmental boundaries such that obstacles, while being detected, are within the robot's boundaries.

5.2. *Team B – the EOD robot designers*

Members of Team B – the EOD robot designers – do not have well-defined guidelines for their design and struggled to generate a detailed process model. Team B's models, instead, consider the design problem from a very abstract perspective and only include the key flows. Team B, like Team A, generates a black box model; their overall goal is to disarm an explosive ordinance. The black box model is decomposed into four key events: setup robot, manoeuvre robot, capture bomb and disarm bomb. The event model, shown as Figure 13, includes a time line to represent the amount of time their customer has allotted for disarming an explosive ordinance.

Specific details such as the robot's transport, operating environment and operational hazards that could be used to define the specific configurations for the EOD robot are not available to Team B. Without this information, Team B is unable to make informed decisions about configuration requirements, and they decide to forgo the generation of configuration models. Team B instead generates sub-event models or more detailed event models for each event in Figure 13. These models are generated by first considering each event in Figure 13 as a black box and then generating its decomposition as a chain of high-level actions that the robot must complete during an event. An example sub-event chain for the capture bomb event is provided in Figure 14. The decomposition of the capture bomb event includes four sub-events: stop robot, move arm, grab bomb and secure bomb.

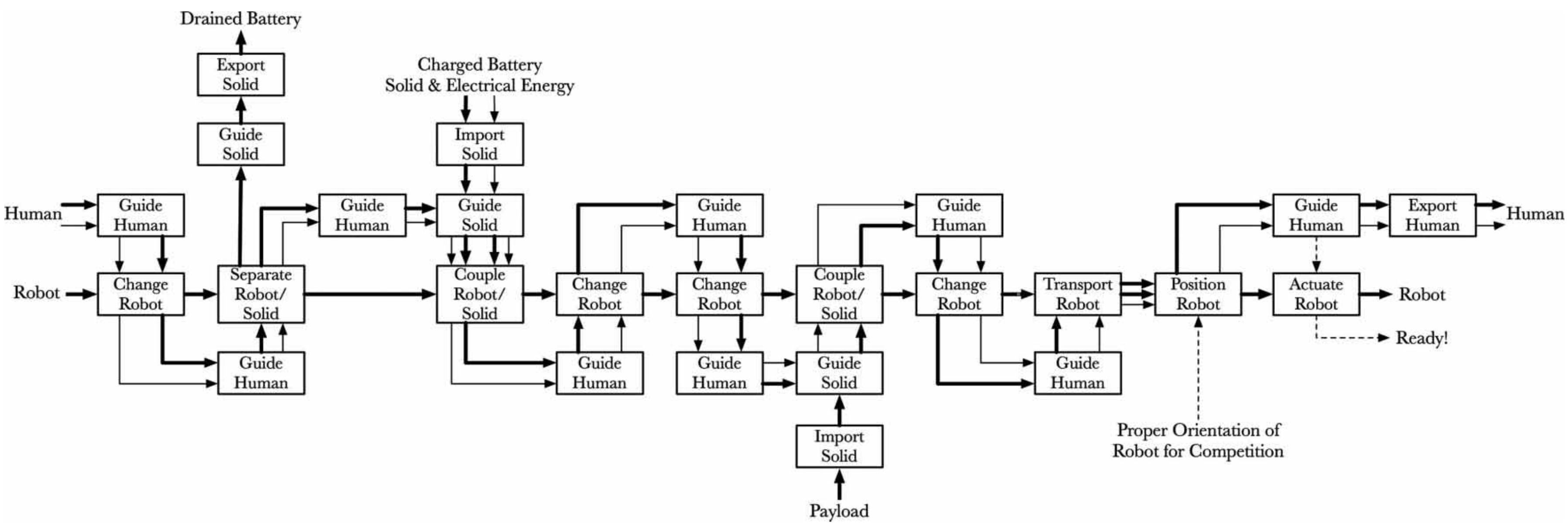

Figure 12. Configuration model generated by Team A for the event of setting up the intelligent ground vehicle.

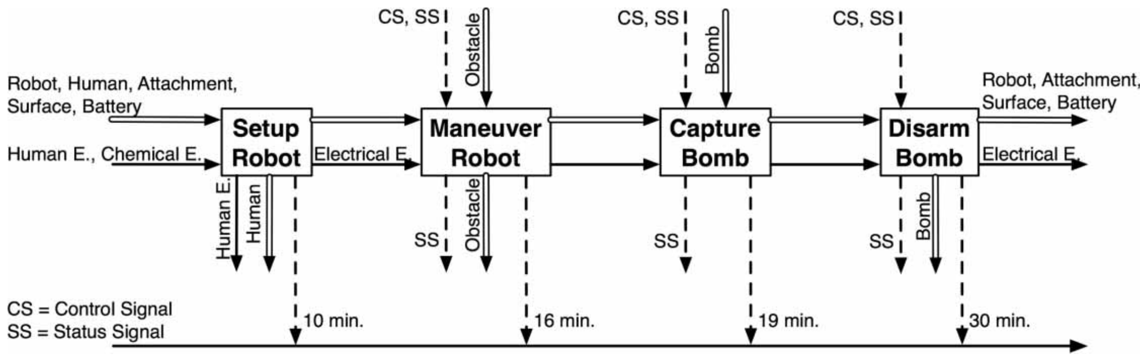

Figure 13. Event model generated by Team B for the process of neutralising an explosive ordinance.

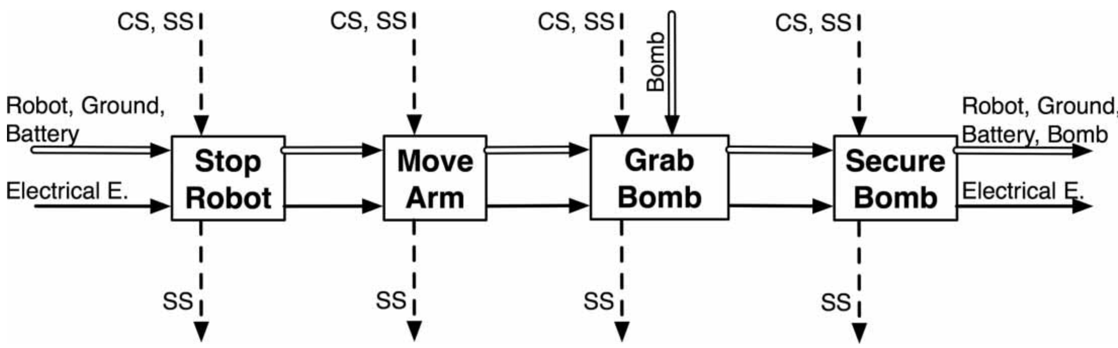

Figure 14. Sub-event model generated by Team B for the event capture bomb.

Following the creation of their process models, Team B, like Team A, iterates the functional model for their EOD robot to include any new insights ascertained. Like Team A, the process analysis leads Team B to more carefully consider how to define their product's boundary and what to include within their environmental boundary. Also, from the setup event, Team B realises that if their robot is to be a modular design (as is requested by the few customer needs available to them), the robot needs to be represented by multiple flows in the event models. By splitting their robot into multiple flows based on product modules, the robots different configurations (based on combinations of unique product modules) could be modelled in either events or configurations by combining individual modules to create 'new' products based on specific applications. The robot, as a modular design, might then be considered a system of systems where the product itself is a system of modules, and each of the modules may too be considered a system. As such, each module could then be modelled with a single functional model, and the process models could then represent how each module fits together for different applications of the robot.

5.3. *Discussion on the case study – process models in conceptual design*

While process and event modelling can certainly be employed as a stand-alone methodology, these case studies indicate benefit from the application of process modelling in conjunction with functional modelling during conceptual design. Discussions with members of both teams indicate that the application of process modelling is useful to scope their specific design problems. Both teams found, while generating their process models, that they had not previously completely considered how their customer would utilise the product during their functional analysis. Utilising what was learned from the process models, both teams iterated their functional models to better define the product and environmental boundaries to more accurately represent the expected operations of their robots.

Process modelling applied in conjunction with functional modelling may be used to extend the traditional single-configuration functional model of a product to a series of functional models representing separate product modules and/or configurations. Team B alluded to this with their discussion of explicit representation of the modularised EOD robot through the process models. Following this, traditional functional models could be created for each event in the process model to represent changes in functionality as the product transitions between configurations or events in a process. Exploration into these applications of process modelling is a part of the researchers' future work.

From the user-defined levels of fidelity with the process models in the two case studies, it might also be concluded that rigidly adhering to method-defined levels of abstraction is not necessary to ascertain benefit from the overall process modelling methodology presented in this paper. This conclusion is consistent with the general application of abstractions in engineering design, where Andreasen (1994) in *Modeling – The Language of the Designer* has this to say on the topic, 'It is important that the product developer realizes which properties are to be modelled and for what purpose. It is also important to limit the modelling, so that not all types of questions are demanded from one model'. Or, more specifically with functional modelling, Ulrich and Eppinger (2004) have this to say, 'There is no single correct way of creating a function diagram and no single correct functional decomposition of a product'. Abstractions are a tool, by which engineers answer questions. 'Abstraction allows us [designers] to consider a greater range of possibilities in which the problem can be dissected into parts or sub-problems, together with ways in which solutions to these sub-problems can be coupled to form complete design solutions' (Voland 2004). Thus, it is necessary for an abstraction technique, such as process modelling, to provide the tools within its methodological structure for generating abstractions at varying levels of user-defined fidelity. All these tools (i.e. levels of model fidelity), however, may not be applicable to every design problem.

6. Conclusion

The process modelling approach presented in this paper provides an abstraction procedure that adds breadth to traditional functional modelling such that models focus beyond the boundaries of the product on how or where the product may be used. Applying process modelling in conjunction with functional modelling during the design process will provide many benefits to engineers during the conceptual design. First, where traditional product focused design methodologies risk inadequate consideration of environment-product effects, the process modelling methodology presented here considers environment interactions as a key element of product design. Second, where traditional product focused design techniques tend to focus on a single configuration of a product or an aggregation of configurations, the proposed process modelling method allows for individual configurations to be modelled together or independently. Third, through a newly proposed time line, events and configurations can be mapped to a temporal domain, providing project-scheduling-like benefits to product design. Fourth, the utilisation of functional modelling techniques provides guidelines for clearer configuration modelling and allows the hierarchical relationship between the product decomposition and its application to be represented. Specified functions and flows maintain clarity through process events, increase efficiency and ensure uniform results. This combination of features pulls together the best features of project-scheduling, activity and workflow models, process mapping, flowcharting and functional modelling techniques to develop a simple yet effective process modelling technique.

To further explore the application of process modelling during conceptual design, process modelling will be explored as a facilitator in the collection of customers' needs to allow designers to investigate design problems following an outcome-driven design paradigm (Ulwick 2005).

Once process models of the customers' needs are generated, the models can be investigated for potential weaknesses in the customers' current actions. Places where weaknesses exist in the customers' current actions represent ideal locations where assistive technologies could be utilised to allow the customer to more effectively reach their desired outcome. Integrated functional and process modelling would thus be generated very early in the design process and be iterated as the design evolves. Therefore, instead of the design-based abstractions focusing on how the product will include each feature, it will focus on the outcomes expected by the customers as they apply each feature in the design.

References

Andersson, J., Pohl, J., and Eppinger, S.D., 1998. A design process modeling approach incorporating nonlinear elements. *In: ASME design theory and methodology conference*, 13–16 September 1998, Atlanta, GA. New York: ASME.

Andreasen, M.M., 1994. Modelling – the language of the designer. *Journal of Engineering Design*, 5 (2), 103–115.

Cho, S.-H. and Eppinger, S.D., 2001. Product development process modeling using advanced simulation. *In: ASME design theory and methodology conference*, 9–12 September 2001, Pittsburgh, PA. New York: ASME.

Christensen, C.M. and Raynor, M.E., 2003. *The innovator's solution: generating and sustaining successful growth*. Boston: Harvard Business School Press.

Cross, N., 2000. *Engineering design methods: strategies for product design*. Chichester, UK: John Wiley and Sons.

Cutherell, D., 1996. Chapter 16: product architecture. *In*: M.D. Rosenau Jr., ed. *The PDMA handbook of new product development*. New York: Wiley, 217–236.

Darnton, G. and Darnton, M., 1997. *Business process analysis*. London: International Thomson Business Press.

Dieter, G., 1991. *Engineering design: a materials and processing approach*. 2nd ed. New York: McGraw-Hill.

Eder, J. and Liebhard, W., 1995. The workflow activity model WAMO. *In: The 3rd international conference on cooperative information systems* (CoopIS), 9–12 May 1995, Vienna, Austria. 87–98.

Fagerström, B. and Nilsson, P., 2003. Integrated product and process modeling. *In: ASME design theory and methodology conference*, 2–6 September 2003, Chicago, IL. New York: ASME.

Friedenthal, S., Moore, A., and Steiner, R., 2008. *A practical guide to SysML: the systems modeling language*. Burlington, MA: Morgan Kaufmann Publishers.

Gero, J.S., 1990. Design prototypes: a knowledge representation schema for design. *AI Magazine*, 11 (4), 26–36.

Goel, A.K., Rugaber, S., and Vattam, S., 2009. Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23 (1), 23–35.

Goslin, L.N., 1967. *The product-planning system*. Homewood, IL: Richard D. Irwin.

Hipple, E.V., 1988. *The sources of innovation*. New York: Oxford University Press.

Hundal, M., 1990. A systematic method for developing function structures, solutions and concept variants. *Mechanism and Machine Theory*, 25 (3), 243–256.

Hunt, V.D., 1996. *Process mapping*. New York: John Wiley and Sons.

Hutcheson, R.S., *et al*., 2006. A function-based methodology for analyzing critical events. *In: ASME design automation conference*, 10–13 September 2006, Philadelphia, PA. New York: ASME.

Ju, T.L., 2001. Process definition for adaptable workflow management systems [online]. *In: XML Europe conference*, 21–25 May 2001, Berlin, Germany. Alexandria, VA: GCA. Available from: http://www.gca.org/papers/xmleurope2001/papers/html/s15-1.html [Accessed 13 January 2006].

Kelley, J.E. and Walker, M.R., 1959. Critical-path planning and scheduling. *In: Proceedings of the eastern joint computer conference*, 1–3 December 1959, Boston, MA. New York: ACM, 160–173.

LeBoeuf, M., 1987. *How to win customers and keep them for life*. New York: Berkley Books.

Malcolm, D.G., *et al*., 1959. Application of a technique for research and development program evaluation. *Operations Research*, 7 (5), 646–669.

Mayer, R.J., *et al*., 1995. *Information integration for concurrent engineering (IICE) IDEF3 process description capture method report* [online]. College Station, TX: Knowledge Based Systems, Inc. Available from: http://www.idef.com/pdf/Idef3_fn.pdf [Accessed 19 October 2007].

Miles, L., 1961. *Techniques of value analysis and engineering*. New York: McGraw-Hill.

Nagel, R.L., Stone, R.B., and McAdams, D.A., 2006. A process modeling methodology for automation of manual and time dependent processes. *In: ASME design for manufacturing and the lifecycle conference*, 10–13 September 2006, Philadelphia, PA. New York: ASME.

NIST (National Institute of Standards and Technology), 1993. *Integration definition for function modeling (IDEF0)* (FIPS Publication 183) [online]. Available from: http://www.itl.nist.gov/fipspubs/idef02.doc [Accessed 15 January 2006].

OMG (Object Management Group), 2009. *Unified modeling language – UML* [online]. Available from: http://www.uml.org/ [Accessed 7 April 2009].

Otto, K. and Wood, K., 2001. *Product design: techniques in reverse engineering, systematic design, and new product development*. New York: Prentice-Hall.

Pahl, G., *et al*., 2007. *Engineering design: a systematic approach*. 3rd ed. London: Springer Verlag.

Randolph, W.A. and Posner, B.Z., 1988. *Effective project planning and management: getting the job done*. Englewood Cliffs, NJ: Prentice Hall.

Rosenau, M.D.J. and Githens, G.D., 2005. *Successful project management: a step-by-step approach with practical examples*. Hoboken, NJ: John Wiley and Sons.

Spinner, M.P., 1997. *Project management principles and practices*. Upper Saddle River, NJ: Prentice Hall.

Stone, R., Wood, K., and Crawford, R., 2000. Using quantitative functional models to develop product architectures. *Design Studies*, 21 (3), 239–260.

Suh, N., 1990. *The principles of design*. New York: Oxford University Press.

The New Oxford American Dictionary, 2005. Oxford: Oxford University Press.

Tor, S.B., *et al*., 2002. Guiding functional design through rule-based causal behavior reasoning. *International Journal of Production Research*, 40 (3), 667–682.

Ullman, D.G., 2002. *The mechanical design process*. New York: McGraw-Hill.

Umeda, Y., *et al*., 1990. Function, behaviour, and structure. *In*: J.S. Gero, ed. *Applications of artificial intelligence in engineering V.* Berlin: Springer Verlag, 177–193.

Ulrich, K.T. and Eppinger, S.D., 2004. *Product design and development*. 3rd ed. Boston, MA: McGraw-Hill/Irwin.

Ulwick, A.W., 2005. *What customers want: using outcome-driven innovation to create breakthrough products and services*. New York: McGraw-Hill.

US Office of Personnel Management, 2001. *5 Step workforce planning model* [online]. Available from: http://www.opm.gov/workforceplanning/wfpmodel.htm [Accessed 13 January 2006].

Voland, G., 2004. *Engineering by design*. 2nd ed. Upper Saddle River, NJ: Pearson Prentice Hall.

Wallace, D.R., Abrahamson, S.M., and Borland, N.P., 1999. Design process elicitation through the evaluation of integrated model structures. *In: ASME design theory and methodology conference*, 12–16 September 1999. Las Vegas, NV: ASME.

Webster's Encyclopedic Unabridged Dictionary of the English Language, 2001. Vol. 8. New York: Random House Value Publishing, 1235–1236, 1357–1358.

Westland, J., 2006. *The project management life cycle*. London: Kogan Page.

Williams, P.B., 1996. *Getting a project done on time*. New York: Amacon.

Zhang, W.Y., *et al*., 2002. Functional design of mechanical products based on beavior-driven function-environment-structure modeling framework. *Innovation in Manufacturing Systems (IMST) on DSpace@MIT* [online]. Available from: http://hdl.handle.net/1721.1/4031 [Accessed 3 March 2009].