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

THE FUNCTIONAL MODELLING ACCOUNT OF STONE AND WOOD: SOME CRITICAL REMARKS

Pieter E. Vermaas
Delft University of Technology, the Netherlands

ABSTRACT

In this paper I review the account of functional modelling in engineering design as proposed by Robert Stone and Kristin Wood in 2000. This account allows overall product functions to be modelled as sets of connected elementary subfunctions. The product functions are described in verb-object forms and represented by black-boxed operations on flows of materials, energies and signals. Subfunctions are described in verb-object forms as well but represented by well-defined basic operations on well-defined basic flows of materials, energies and signals, as laid down in libraries, which are called a functional basis. The review aims at analysing the internal structure of the account, with the methodological means as provided by analytic philosophy. Firstly I demonstrate that the account harbours two notions of functions: one described by verb-object forms and one by verbs only. Secondly I argue that overall product functions and subfunctions described by verb-object forms need not always coincide with what is represented by operations on flows; operations-on-flows descriptions as used by Stone and Wood may better be understood as representing behaviour of products and their components, and product functions and subfunctions may better be represented by those parts of the operations-on-flows descriptions that concern flows that are actually intended by designers. Thirdly I argue that subfunctions as defined in the account need not always coincide with subfunctions that can be taken as elementary functions in designing, and discuss alternative possibilities for accommodating these elementary design functions in the account of functional modelling.

Keywords: Functions, functional modelling, functional basis, functional decomposition, behaviour

1 INTRODUCTION

In this paper I review the account of functional modelling in engineering design as proposed by Robert Stone and Kristin Wood in 2000 [1]. Briefly put, this account allows overall product functions from especially the electromechanical and mechanical domain to be modelled as sets of connected elementary subfunctions. In line with the design methodology of Pahl and Beitz [2] an overall product function is described in a verb-object form and represented by a black-boxed operation on flows of materials, energies and signals. A subfunction is also described in a verb-object form but represented by a well-defined basic operation on well-defined basic flows of materials, energies and signals. The black-boxed operations on general flows representing product functions are derived from costumer needs, and the basic operations and basic flows representing subfunctions are laid down in common and limited libraries that span the functional design space. These libraries are called a *functional basis*. Stone and Wood present their account as supporting a number of engineering tasks, including the archiving, comparison and communication of functional descriptions of existing products, and the engineering designing of new products.

Since it has been proposed, Stone and Wood have with a number of collaborators successfully developed the account. In 2002 the functional basis libraries have been slightly modified together with Julie Hirtz, Daniel McAdams and Simon Szykman [3] by integrating the original libraries proposed in [1] with similar libraries developed at the US National Institute of Standards and Technology [4]. A web-based repository storing functional models of up to 102 products has been created at the Design Engineering Lab of the University of Missouri-Rolla [5], and an automated mathematically-based design tool called the Concept Generator is developed [6], underlining the usefulness of the proposal for the mentioned tasks of archiving and designing, respectively. The account is currently even applied outside engineering proper, for modelling also functionally processes, manual operations and human-centric procedures [7].

The success of the account calls for reviewing functional modelling, reviewing that assesses the technological scope and usefulness of the account, and reviewing that focuses on its internal structure. The review given in this paper is one of the latter sort. In section 2 I give a brief overview of the account, considering the basic concepts in terms of which the account is formulated and introducing the tasks it is taken to be useful for. This review focuses on Stone and Wood's original paper [1] and partly on the later paper on the Concept Generator [6]. The functional basis libraries are adopted from [3]. In section 3 I switch to a more critical mode. It is demonstrated that the account harbours two notions of functions: one described by verb-object forms and one by verbs only. Then it is argued that overall product functions and subfunctions described by verb-object forms need not always coincide with that what is represented by operations on flows of materials, energies and signals. Finally it is argued that subfunctions as defined in the account need not always coincide with subfunctions that can be taken as elementary in designing, raising some problems for the claim that the account of Stone and Wood facilitates designing. (The critical remarks do not, by and large, affect the claim that the account also facilitates archiving). Section 4 continues in a more constructive mode. Firstly I argue that operations-on-flows descriptions may better be understood as representing behaviour of products and their components, and propose to represent product functions and subfunctions by the parts of the representations of behaviour that concerns flows that are actually intended by designers. Secondly I discuss possibilities in which functional modelling can deal with the distinction between subfunctions defined by the functional basis and subfunctions that can be considered as elementary in designing.

Before starting I briefly say something about the methodology adopted. The review has its origin in analytic philosophy of engineering, meaning that the prime methods adopted are analysis of concepts and assumptions by means of argument and by means of what may be seen as "toy examples." Because of these methods the review may seem critical and academic, failing to be of direct use to furthering functional modelling from an engineering point of view. I accept this disadvantage. The aim of the review is to understand what is proposed; the account by Stone and Wood is of great interest, also for philosophers working on engineering. The reason to present the review also to an engineering audience is an expectation that it may be of use for an engineering evaluation of the account. Yet, given my analytic methods the contribution to possible improvements will be limited to suggestions, leaving authority with the experts. Engineers and philosophers have different methods and aims, but start to collaborate increasingly as exemplified by, for instance, the engineering field of formal ontology (e.g., [8]). This collaboration may be of worth to both, but can lead to interdisciplinary clashes and irritation as they occur also within engineering proper [9].

2 THE FUNCTIONAL MODELLING ACCOUNT OF STONE AND WOOD

2.1 Functional modelling

The functional modelling account of Stone and Wood [1] allows, as said, to model overall product functions as sets of connected subfunctions. This modelling is presented as supporting the archiving, comparison and communication of functional descriptions of existing products, as well as the engineering designing of new products. Archiving, comparison and communication is assisted since the subfunctions into which overall product functions are decomposed, are described in a common universal language. Designing of new products is supported since functional modelling allows designers to make critical design decisions about the product's architecture in the early conceptual stage of designing at which only functional descriptions are considered. Moreover, functional modelling provides the means to decompose overall product functions into what Stone and Wood call 'small, easily solvable subfunctions', that is, subfunctions for which solutions exist such that 'the [structural] form of the [product to be designed] follows from the assembly of all sub-function solutions' ([1], section 3.1). Stone and Wood focus primarily on this second goal and introduce their account in terms of the tasks designers have to carry out when applying the account. A course-grained description of these tasks as presented in [1], section 5, is as follows.

The first task is to arrive at an overall product function of a product to be designed, described in a verb-object form and represented by a black-boxed operation on flows of materials, energies and signals (see Figure 1). This black-boxed operation originates from customer needs and may initially be quite general and is refined later on in the design process. Thus, in terms of one of Stone and Wood’s examples of an electrical hot air popcorn popper, initially the input flow of materials contains corn kernels, the input flow of energies contains electricity, the output flow of material contains popcorn, and the rest of the flows are left unspecified. Later on the input is defined as corn kernels, butter, air and electrical energy, and the output as popcorn, melted butter, air, thermal energy and pneumatic energy (see Figure 2, adopted from [1], figure 7). The verb-object description of the product function and this (black-boxed) operation-on-flows representation are related in the sense that the verb corresponds with the operation and the object corresponds with (parts of) the flows: the function of the popcorn popper is ‘popping corn’.

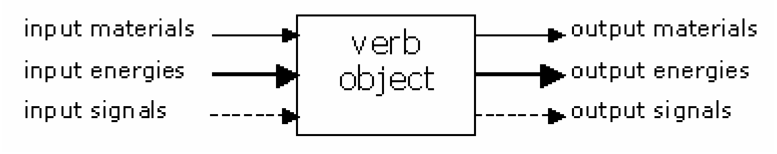

Figure 1. Representation of a product function or subfunction

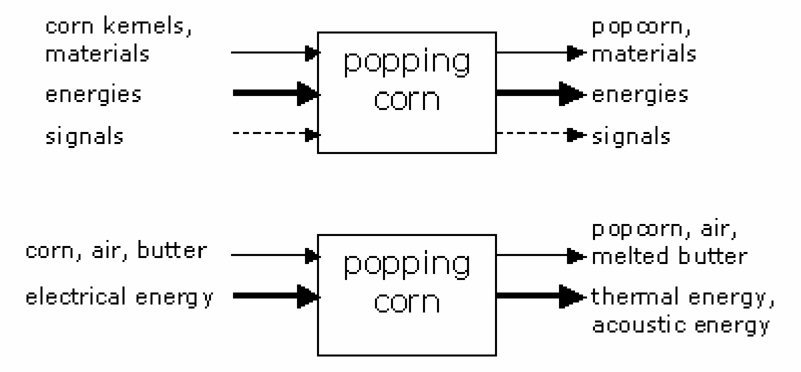

Figure 2. Initial and final overall product function of a popcorn popper

The second task is to define for each input flow a chain of subfunctions that transform that flow step-by-step into an output flow. These subfunctions are also described in verb-object forms and represented by operations on flows. But now the verbs are to be chosen from a fixed library of basic functions and the flows are to be chosen from a fixed library of basic flows. These two libraries make up the functional basis and are listed in Tables 1 and 2 respectively (they are the improved libraries given in [3]). The subfunctions part of the different chains must be ordered in time with respect to one another.

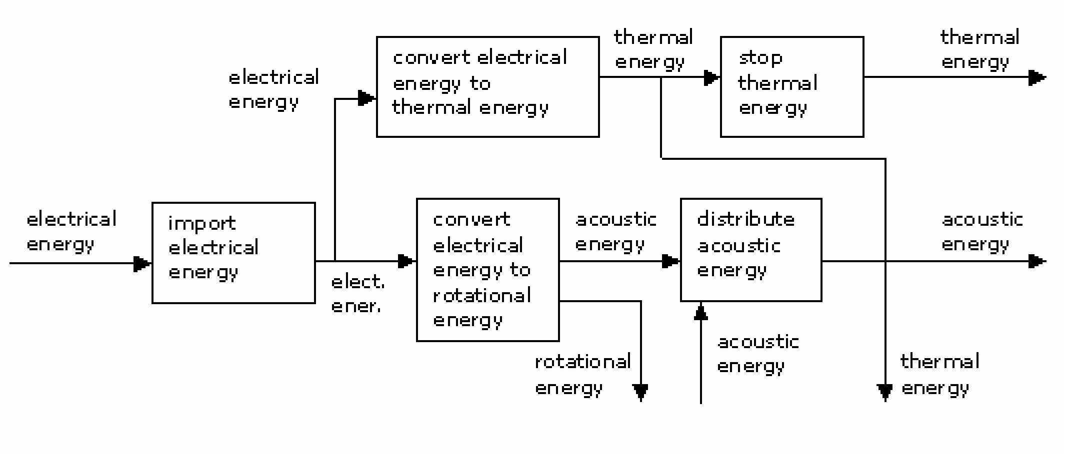

Figure 3. Part of a functional model for Popping Corn (after [1], figure 7)

The third task is that these temporally ordered chains of subfunctions are integrated by connecting the chains, thus arriving at the functional model. For the popcorn popper a part of its functional model is given in Figure 3 (it is the energy-flow part, adopted from [1], figure 7, with verb-object names adjusted to the improved functional basis of [3]; the energy arrows at the right-hand side pointing to the right are output flows, the energy arrows at the bottom pointing downwards and upwards are internal flows to and from the other chains of subfunctions). Figure 4 (adopted from [6], figure 1) gives a full functional model, now for the overall product function 'hold liquid and retain heat' of a cup.

The designer thus has to analyse the input flows of the overall function in terms of basic flows from the library of basic flows, and come up with a series of operations from the library of basic functions that sequentially and/or in parallel transform the input flows step-by-step into basic flows that, together, make up the output flows.

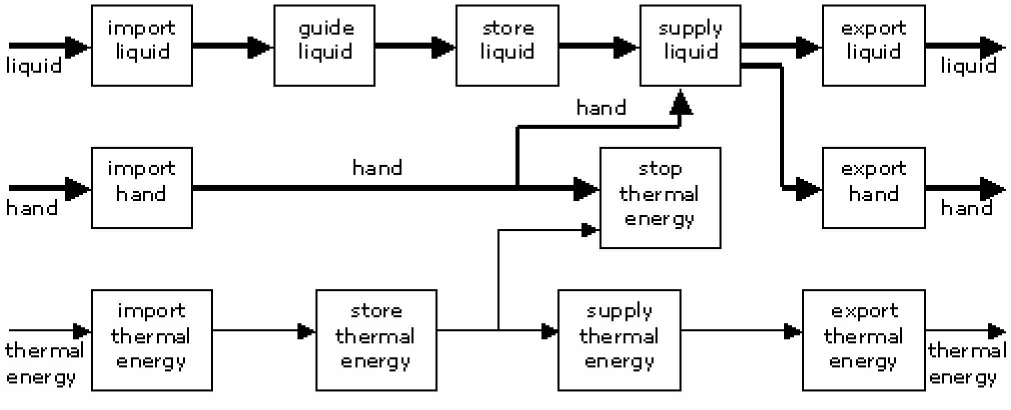

Figure 4. A functional model for 'holding liquid and retain heat' (after [6], figure 1)

2.2 Automated functional modelling

Further examples of functional models of product functions can be found at a web-based repository [5]. This repository stores the functional models of up to 102 existing products, and stores their components as design solutions of the various subfunctions part of these models. At this site one can moreover find an automated mathematically-based design tool called the Concept Generator. This tool is aimed at creating new functional models for any overall product function that is fed into it, and at generating design solutions for these overall product functions on the basis of the design solutions of subfunctions that are already stored in the repository. The Concept Generator solves a part of one of the problems with the functional modelling account that are introduced in the next section. I therefore introduce the algorithm used by the Concept Generator as discussed in [6], section 3. This introduction is again course-grained, focussing on what the algorithm does, and ignoring the means by which it achieves so.

In the first step of the algorithm, the Concept Generator translates the functional model of the overall product function (only models consisting of single chains of subfunctions are considered in [6]) into information about the subfunctions and their adjacency. Secondly, the Concept Generator collects for each individual subfunction in the chain design solutions consisting of components that are stored in the repository as having that specific subfunction.

In the third step all design solutions for the product as a whole are generated by describing on the basis of the information gathered in the first two steps all theoretically possible component chains that solve the overall product function. Fourthly, additional information is collected from the repository on which sets of components have been actually combined in existing products, and in that sense can be taken as sets of compatible components.

Finally, in a fifth step, this information about the compatibility of components is used to prune the set of theoretically possible component chains to a set of feasible component chains that solve the overall product function. In this final step also a ranking is added to these feasible chains, to "bubble the most promising solutions to the top." I here ignore how this ranking is produced; the point relevant for the review is that the algorithm puts constraints on the functional modelling of overall product functions: the algorithm selects only those functional models that consist of subfunctions for which there are components available in the design repository that have these subfunctions, and the algorithm selects only those models that combine subfunctions that are actually combined in functional models of products stored in the repository.

Table 1. Library of basic functions according to [3]

Primary functions	Secondary functions	Tertiary functions
Branch	Separate	
		Divide
		Extract
		Remove
	Distribute	
Channel	Import	
	Export	
	Transfer	
		Transport
		Transmit
	Guide	
		Translate
		Rotate
		Allow DOF
Connect	Couple	
		Join
		Link
	Mix	
Control magnitude	Actuate	
	Regulate	
		Increase
		Decrease
	Change	
		Increment
		Decrement
		Shape
		Condition
	Stop	
		Prevent
		Inhibit
Convert	Convert	
Provision	Store	
		Contain
		Collect
	Supply	
Signal	Sense	
		Detect
		Measure
	Indicate	
		Track
		Display
	Process	
Support	Stabilize	
	Secure	
	Position	

Table 2. Library of basic flows according to [3]

Primary flows	Secondary flows	Tertiary flows
Material	Human	
	Gas	
	Liquid	
	Solid	Object
		Particulate
		Composite
	Plasma	
	Mixture	Gas-gas
		Liquid-liquid
		Solid-solid
		Solid-liquid
		Liquid-gas
		Solid-gas
		Solid-liquid-gas
		Colloidal
Signal	Status	Auditory
		Olfactory
		Tactile
		Taste
		Visual
	Control	Analog
		Discrete
Energy	Human	
	Acoustic	
	Biological	
	Chemical	
	Electrical	
	Electromagnetic	Optical
		Solar
	Hydraulic	
	Magnetic	
	Mechanical	Rotational
		Translational
	Pneumatic	
	Radioactive/Nuclear	
	Thermal	

3 SOME CRITICAL REMARKS

3.1 Product functions and subfunctions versus functions and basic functions

A first remark about the functional modelling account is about terminology, and concerns the term ‘function’ itself. Stone and Wood use this term in two ways. Strictly speaking they define ‘function’ as “a description of an operation to be performed by a device or artifact, expressed by the active verb of the sub-function” ([1], section 2). Basic functions as defined by the functional basis (see Table 1) are thus functions, for also basic functions are descriptions of operations. But product functions and the subfunctions mentioned in the definition of ‘function’ are not functions in this sense. Product functions and subfunctions are also clearly defined by Stone and Wood ([1], section 2), and according to these definitions they are expressed by verb-object forms, capturing the tasks of products or of their components, respectively. Product functions and subfunctions are thus not merely operations, but represented by operations-on-flows. These different senses of the term ‘function’ may lead to confusing statements – a product function is not a function of the product; subfunctions are not basic functions – but Stone and Wood fortunately avoid such confusion. Yet, when separating these two senses, one can start doubting whether the subfunctions that are defined by the functional basis are actually all “small, easily solved subfunctions”. The basic functions given in Table 1 may all look small and easily solvable: for each of the terms that occur in that table – branch, connect, channel, and so on – a quick design solution may come to mind. But these basic functions are merely operations; the subfunctions into which overall product functions are decomposed by functional modelling are not *operations* but *operations-on-basic-flows*. The functional basis generates a lot of such *operations-on-basic-flows* subfunctions and they are not all small and easily solvable. Consider, for instance, the combinations convert-hydraulic-energy-to-nuclear-energy, or join-plasma-and-object. In subsection 3.3 I elaborate on this line of criticism.

3.2 Product functions and subfunctions versus operations-on-flows descriptions

A second and more critical remark concerns the way in which Stone and Wood represent product functions and subfunctions. As said they are expressed by verb-object forms, reflecting the tasks products or their components have to do for customers. And they are represented by descriptions of operations these products or components, respectively, perform on flows of materials, energies and signals, where the operations correspond to the verbs, and the flows to the objects. It can, however, be argued that this representation need not always be accurate. The argument that there are product functions and subfunctions that are not properly represented by operations-on-flows descriptions, may be less convincing. The argument that there are operations-on-flows descriptions that do not always represent product functions or subfunctions is easier to make.

Consider, to give the first argument, products that have the tasks of scanning states-of-affairs, such as radar installations. The product functions of such scanners are accurately expressed by the verb-object forms ‘scanning state-of-affairs’; radar has the product function ‘detecting planes’. If these functions are to be represented by operations-on-flows descriptions, where the operations correspond to the verb ‘scan,’ and the flows to the object ‘state-of-affairs,’ then the flow through the scanners has to include these states-of-affairs. Yet, whether the flow through the scanners actually does include these states-of-affairs, depends on how you precisely choose the system boundaries of the scanners: if a radar installation consists of the hardware of the installation only, planes do not enter and leave radar; the input flow for radar is then electrical energy and the output is light dots on a screen that counts as signals. Hence on this narrow choice of the system boundaries, the operations-on-flows description that represents the radar’s product function is ‘convert electrical energy to optical energy’ which is not an operation on planes. Only if one chooses the system boundary of scanners such that it includes the area scanned – radar installations are then huge devices – can one maintain that the flows through these scanners include the states-of-affairs scanned. And only if this latter choice is acceptable, it can be maintained that the product functions of scanners are represented by operations-on-flows descriptions where the flows include the objects part of the verb-object forms that express these product functions.

The second argument is established when an example can be given of an operations-on-flows description that does not represent a product function or subfunction. This example can be found in the functional model that Stone and Wood give of the hot air popcorn popper. That model (see Figure 3) contains an element called ‘distribute acoustic energy’ (the verb-object description of this element is actually ‘dissipate acoustic energy’ in [1] but in the functional basis of [3] ‘dissipate’ became ‘distribute’). It can, however, be doubted if this operation-on-a-flow represents always a subfunction that is part of the overall product function of the popcorn popper. Assume, for instance, that the customers and designers concerned do not have specific wishes concerning the noise the popcorn popper produces. The acoustic energy flow that arises within the functional modelling is then rather an unintended noise, and the distribution of this noise is not corresponding to a task set by someone.

This point can be put more general. The overall product function of a product such as a popcorn popper (see Figure 2), can be properly represented by the intended flows only: costumers may need a popcorn popper that only transforms corn kernels and electrical energy into popcorn, without caring about the additional output flows of acoustic and thermal energies of the popper. Yet what happens in functional modelling is that this by customers intended operations-on-flows-representation is extended to a detailed operations-on-flows description that includes additional flows. Some of these flows may be necessary from a technical point of view, making them intended as well but now by the designers: there needs to be a flow of air through parts of the popcorn popper. Yet, others of those additional flows seem both for customers and designers unintended by-products: that air, melted butter, thermal energy and acoustic energy are leaving the popper may be unavoidable but eventually unimportant for both customers and designers. These latter unintended flows part of the extended description are then not representing parts of the by the customers or designers required product function of popcorn popping.

3.3 Subfunctions versus small, easily solvable subfunctions

A third remark concerns Stone and Wood's claim that their functional basis defines subfunctions that are "small, easily solvable subfunction[s]" in designing. Again one can create examples that prove that the set of subfunctions as defined by the functional basis does not coincide with the set of product functions that are small and easily solvable for designers. In subsection 3.1 two subfunctions were defined with the functional basis – 'convert-hydraulic-energy-to-nuclear-energy' and 'join-plasma-and-object' – for which it is hard to defend that designers readily know the solution. An example of a product function that is small and easily solvable for designers but not a subfunction by the functional basis is the 'holding liquid and retain heat' example given in section 2 (see Figure 4). This product function is easily solvable – a cup will do the trick – but is not a subfunction since on the account of Stone and Wood it is made up of twelve subfunctions.

More generally one can argue that the notion of a small, easily solvable subfunction and the notion of a subfunction as defined by the functional basis of Stone and Wood, cannot be the same since the first is dependent on the historical and design context considered, whereas the second notion is independent of such contexts. Consider, for instance, the functional-basis subfunction convert-electrical-energy-to-rotational-energy. For designers currently working on, say, household appliances, this subfunction is indeed a small and easily solvable subfunction: electrical motors are nowadays readily available, meaning that the designers need not be bothered about solving this subfunction. But for designers working two hundred years ago this subfunction was rather one that had no solution. Moreover, even today there may exist contexts where this functional-basis subfunction will not be considered as easily solvable. Consider, for instance, designers working on a new type of electrical motor. They will not be helped by taking convert-electrical-energy-to-rotational-energy as an easily solvable one; such designers are rather facing the task of breaking up this functional-basis subfunction into other subfunctions in order to advance with their work. This context-dependence is, however, lost if easily solvable subfunctions are identified with the subfunctions defined by the functional basis of Stone and Wood; the set of functional basis-subfunctions has a fixed content over history and over all engineering contexts.

4 SOME MORE CONSTRUCTIVE REMARKS

4.1 Operations-on-flows-descriptions as representing behaviour

If the remark is accepted that the detailed operations-on-flows-descriptions generated by the Stone-Wood account include unintended basis flows and unintended operations, and thus do not always represent product functions and subfunctions, then two questions arise. The first is what these detailed operation-on-flows descriptions do represent. The second is how to recover from these detailed descriptions the ones that still can be taken as representing product functions and subfunctions. The answer to the first question may be that the detailed operations-on-flows-descriptions represent the *behaviour* of products and components, as has been noted also in [10], section 5, and [11]. Behaviour of products and components is in some design methodologies a key-concept by which designers relate products functions and subfunctions to the physical structure of the products and their components, respectively (e.g., [12], [13], [14]): by their structure, products and components can exercise specific behaviours, of which some are singled out as their product functions and subfunctions, respectively. This answer to the first question also provides means to answer the second: if product functions and subfunctions single out specific behaviours, then these product functions and subfunctions can be taken as being represented by parts of the detailed operations-on-flows descriptions generated by the Stone-Wood account. Consider again the example of the popcorn popper. The initial description of the overall product function of this popcorn popper as given in Figure 2 can be taken as representing its overall product function since it contains those flows costumers consider to be the popper's task. The final description as given in Figure 2 can instead be taken as representing its behaviour since it adds to these intended flows, flows that are also present but not all intended. When the by designers intended flows are taken as representing the product functions and subfunctions, one can represent those functions by highlighting them into the detailed operations-on-flows-descriptions, say by italics and a distinct colour (see Figure 5).

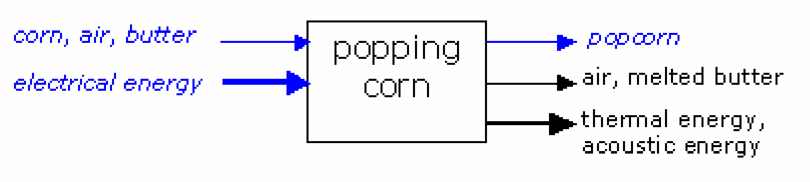

Figure 5. The overall product function as a part of its behaviour

And the operation-on-a-flow-description "distribute acoustic energy" in the functional model of the popcorn popper (Figure 3) can be taken as representing behaviour of a component of the popper, but not representing a subfunction part of the popper's product function since it does not concern an operation-on-a-flow intended by designers.

These answers imply that in the account of functional modelling a distinction is made between flows that are intended by designers, and flows that are unavoidable but not intended. The full, detailed operations-on-flows descriptions represent behaviours, which may include the unintended flows and operations on those flows; the intended parts of these detailed descriptions represent the product functions and subfunctions.

4.2 Adding context-dependent small, easily solvable subfunctions

The problem that functional modelling may yield subfunctions that are not easily solvable, may be avoided by invoking the algorithm of the Concept Generator. By this algorithm (see section 2.2), all the subfunctions that will turn up in feasible functional models are subfunctions for which design solutions are stored in the repository. Hence, a functional model that contains a subfunction such as 'convert-hydraulic-energy-to-nuclear-energy' will not be generated.

Let us accept this response and focus on the problem that functional modelling may conversely decompose a small and easily solvable product function (or subfunction) into a cluster of functional-basis subfunctions. Can the Concept Generator, or extensions thereof, also solve this second problem, for instance, by identifying a single design solution for such a cluster of non-functional-basis subfunctions? I believe it may be defended that the answer is positive, but the consequence of this defence seems to be that the repository should store all possible behaviour of components rather than only their subfunctions. Consider, for instance the example of the easily solvable 'holding liquid and retain heat' product function that in functional modelling is broken up into a cluster of twelve functional-basis subfunction (see Figure 4). The Concept Generator may identify a cup as a single design solution to 'holding liquid and retain heat' if one assumes that the repository stores 'cups' as design solutions to all the twelve functional basis-subfunctions involved, and assumes that the algorithm can rank this single design solution as a highly feasible one. The assumption that 'cups' are indeed stored as design solutions to all the twelve functional basis-subfunctions of Figure 4 may seem reasonable. But consider now the non-functional-basis subfunction 'providing electrical energy and stabilising', which may come up when designing a cellular phone (the 'stabilisation' part of this subfunction may capture the costumer needs that the phone lies nicely in the hand). A design solution to this subfunction may be a battery placed at the lower halve of the phone. But the Concept Generator may identify this solution only if batteries are stored in the repository as a design solution to stabilisation. This assumption seems, however, difficult to make, since batteries may never have been taken as design solutions to stabilisation before the time of the designing of the first cellular phones. The assumption seems tenable only if the repository, in addition to information about the subfunctions of components, contains also information about their behaviour: if the repository contains the information that batteries have relatively high mass densities, then the Concept Generator indeed may find batteries as design solutions to the subfunction of stabilisation.

A more rigorous solution to the problem that the set of functional-basis subfunction need not coincide with the set of small, easily solvable subfunctions, is to add the second set as a separate ingredient to the account of functional modelling. If this is done, the next step becomes to look for an algorithm, such as the one employed in the Concept Generator, that can identify clusters of functional-basis subfunctions that, together, form easily solvable non-functional-basis subfunctions. Functional modelling can in that case amount to a decomposition of overall product functions into functional-basis subfunctions and to a second decomposition in which those overall product functions are more coarse-grained broken up into easily solvable subfunctions. The question of whether such an algorithm exists lies beyond the realm of this review. If, however, the set of functional-basis subfunction is distinguished from the set of small, easily solvable subfunctions, then the functional modelling account of Stone and Wood becomes more sensitive to the tasks designers in different historical and/or engineering contexts face: the account then reproduces that for designers who worked two hundred years ago or who nowadays create modern engines, ‘convert-electrical-energy-to-rotational-energy’ is not an easily solvable subfunction. Moreover, distinguishing these sets may also be beneficial for the other task for which functional modelling is proposed, that is, for the archiving, comparison and communication of functional descriptions of existing products. The functional basis, by giving a fixed, universal and context-independent language for characterising subfunctions, provides a strong common basis to functional descriptions of products; the set of easily solvable subfunctions, by being context-dependent, provides the means to lay down how designers solved their tasks in terms of the means they had.

5 CONCLUSIONS

In this paper I reviewed the account of functional modelling in engineering design as proposed by Robert Stone and Kristin Wood [1]. This account allows overall product functions to be modelled as sets of connected elementary subfunctions. These product functions and subfunctions are described in verb-object forms and represented by black-boxed operations on flows of materials, energies and signals. Subfunction are described in verb-object forms but represented by well-defined basic operations on well-defined basic flows of materials, energies and signals, as laid down by the functional basis [3]. The review aimed at analysing the internal structure of the account, with the methodological means of analytic philosophy. It was demonstrated that the account harbours two notions of functions: one described by verb-object forms and one by verbs only. Then it was argued that overall product functions and subfunctions described by verb-object forms need not coincide with that what is represented by operations on flows. Finally it was argued that subfunctions as defined in the account need not always coincide with subfunctions that can be taken as small and easily solvable.

In the more constructive part of this paper I firstly argued that the operations-on-flows descriptions as used by Stone and Wood may better be understood as representing behaviour of products and of their components, and proposed to represent product functions and subfunctions by the parts of the representations of behaviour that concerns flows that are actually intended by designers. Secondly I discussed possibilities in which functional modelling can deal with the distinction between subfunctions defined by the functional basis and subfunctions that can be considered as easily solvable in designing. The algorithm of the Concept Generator can prevent that product functions are decomposed into non-solvable subfunctions, and similar algorithms may help to identify easily solvable subfunctions consisting of clusters of functional-basis subfunctions. I also briefly discussed the bolder proposal of adding the set of easily solvable subfunctions to the account of functional modelling.

With this last proposal the review has been stretched to its very limits and may have reached a point at which the engineering reader may find it of less value if not vacuous. This possibility is the downside of the methodology adopted for this review. Analytic philosophy focuses on internal conceptual issues. Its strength lies in identifying problems, its weakness is that it is less suited – possibly unsuited – for solving these problems. The review therefore has to stop, and I submit its results for evaluation and as possible starting points to improve on a substantial and viable account of functional modelling.

REFERENCES

[1] Stone, R.B. and Wood K.L. Development of a functional basis for design. *Journal of Mechanical Design*, 2000, 122, 359-370.

[2] Pahl G. and Beitz W. *Engineering Design: A Systematic Approach*, 1996 (Springer, Berlin).

[3] Hirtz J., Stone R.B., McAdams D.A., Szykman S. and Wood K.L. A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design*,

2002, 13, 65-82.

[4] Szykman S, Racz J.W. and Sriram R.D. The representation of funcion in computer-based design. In *Proceedings of the 1999 ASME Design Engineering Technical Conferences (11th International Conference on Design Theory and Methodology), September, 1999, Las Vegas, NV, USA*, paper no. DETC99/DTM-8742, 1999 (ASME).

[5] http://function.basiceng.umr.edu/delabsite/repository.html

[6] Bryant C.R., McAdams D.A., Stone R.B., Kurtoglu T. and Campbell M.I. A validation study of an automated concept generator design tool. In *Proceedings of the 2006 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, PA, USA*, paper no. DETC2006-99489, 2006 (ASME).

[7] Nagel R.L., Stone R.B. and McAdams D.A. A process modeling methodology for automation of manual and time dependent processes. In *Proceedings of the 2006 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, PA, USA*, paper no. DETC2006-99437, 2006 (ASME).

[8] Varzi A.C. and Vieu L. *Formal Ontology in Information Systems: Proceedings of the Third International Conference (FOIS-2004)*, 2004 (IOS Press, Amsterdam).

[9] Bucciarelli L.L. *Designing Engineers*, 1994 (MIT Press, Cambridge, MA).

[10] Chandrasekaran B. Representing function: relating functional representation and functional modeling research streams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2005, 19, 65-74.

[11] Garbacz P. A formal model of functional decomposition. In *Proceedings of the 2006 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, PA, USA*, paper no. DETC2006-99097, 2006 (ASME).

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

[13] Chandrasekaran B. and Josephson J.R. Function in device representation, *Engineering with Computers*, 2000, 16, 162-177.

[14] Vermaas P.E. and Dorst K. On the conceptual framework of John Gero's FBS-model and the prescriptive aims of design methodology. *Design Studies*, 2007, 28, 133-157.

Contact: Pieter E. Vermaas
Delft University of Technology
Department of Philosophy
Jaffalaan 5
NL-2628 BX, Delft
The Netherlands
Phone: + 31 (0)15 2783323
Fax: + 31 (0)15 2786439
e-mail: p.e.vermaas@tudelft.nl
URL: http://www.tbm.tudelft.nl/webstaf/pieterv/