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

FUNCTIONAL VECTOR SPACE

Gualtiero Fantoni[1], Riccardo Apreda[1] and Andrea Bonaccorsi[1]
[1] University of Pisa (Italy)

ABSTRACT

Recent developments in Functional Analysis witness the emergence of a new paradigm. Formal models seem destined to replace the standard taxonomical and descriptive approaches. In this paper we propose a new representation of the function space as a vector space.
After discussing the rationale for the introduction of a formal architecture, we provide a set of rules/guidelines to construct the vector space of functions (actions on flows) and individuate in physical and logical laws the ground for its generating vector basis.
Finally, we discuss some key and unresolved issues and open several avenues for further research in the exciting and fast moving field of formal modelling of function spaces.

Keywords: Functional Analysis, Functional Vector Space, Vector Functional Modelling

1 INTRODUCTION AND STATE OF THE ART

Functional analysis evolved during its almost forty years of life, from the original proposal by Miles [1] and the first rationalization by Pahl and Beitz [2] to the attempt of the NIST [3] and the recent interesting work of Stone et al. [4] finally synthesised by Hirtz et al. [5].
The approach consolidated along the years is based on a precise identification of each electromechanical function that can occur in a technical product or process with a pair of words, a verb indicating an action plus its object, one of the three possible flows of material, signal, and energy [2].
The main research activity in the field has been devoted to explore, catalogue and organize the entire range of all possible functions and flows, often through the painstaking analysis of a huge number of patents and other technical descriptions [6][3]. In organizing the functional databases the criteria followed have been hierarchy and synthesis. The result has been the distillation of restricted set of functions of very general use and importance, of which all other functions can be considered either special cases or synonyms. Such restricted sets (no more than 150 entries), are commonly referred to as Functional Basis, the most recent one being the so-called Reconciled Functional Basis [5], and present a vertical, tree-like structure.
Within such approach, Functional Analysis mainly consists in the description of a product or a process through a Functional Scheme, that is through an ordered list of abstract functions belonging to the Functional Basis, connected by the flows that constitute the input or output of each function.
The present model based on hierarchical Functional Basis and on Functional Schemes descriptions makes Functional Analysis already a very powerful and effective tool. However, even if the academic research in the field is entering in its mature phase, Functional Analysis is not as widespread and used in the design, engineering and productive world as it should, while in our opinion its enormous potentiality are still not fully deployed and exploited. The reasons are both theoretical and practical.
On the practical side, to be really effective and fruitful, Functional Analysis needs a certain degree of mastering that may not be quick to achieve for the novice. Moreover, existing Functional Databases are flawed by internal incoherences and there is no common agreement among them. As well, it is difficult to define clear and objective rules or algorithms on how to use them besides the mere cataloguing aim (see the attempts in [7][8]). Repositories of Functional Schemes are constructed manually, and after the retrieval phase entries have to be used and analysed one by one.
On the theoretical side, the main reason is the absence of a formal theoretical model that could ground and give objectiveness to the Databases and from which the rules to use them in the analysis could be deducted. A solid theoretical frame would also consequently make possible the elaboration of software tools for semi-automatic analysis, a necessary feature in order to boost a current use of the methodology.

We devoted a forthcoming paper to a full-scale analysis of the critical aspects of the standard model for Functional Analysis. In the present one we intend instead to move the first step towards a radically new, fully formal paradigm, able to found the methodology on a solid theoretical ground and setting the roadmap to definitely overcome the classical construction's shortcomings.

Our starting point has been the need to go beyond the misleading identification between functions and verbs of the natural language. The true foundation of functions must be sought into the laws of nature, in the principles of physics, chemistry and logics.

Another limiting scheme that has to be overcome is the tree-like structure of functional databases: the complex nature of functions requires a more sophisticated network of relationships.

A rather natural evolution of these two assumptions is the realization that physical or logical principles may be seen as the generating elements of the ensemble of existing functions, in an analogous way as a vector space is generate by a set of base vectors.

We have hence developed the vector hypothesis and found evidence for the existence of such a representation for the Functional Space. In this context, the rules of linear algebra and geometry could provide a very rich, yet rigorous, framework to implement the different functional relationships.

The vector approach to Functional Analysis leads to a restricted set of fundamental objects, *i.e.* a proper Functional Basis, in an unambiguous way. At the same time, it does not lose the great variety and richness provided by the complete functional world: Functional Vector Basis can be integrated within the extended databases [9] almost by definition, and each element can be easily related to the others and to the basis vectors (and, in the end, to the law of physics). Actually, the hints coming from the vector model constitutes a valid way to improve and harmonize the categorization and the hierarchical structure even of the traditional, tree-like Functional Basis and Databases.

Among the several implications of the new model, we quote here the possibility of introducing <u>quantitative statements and comparisons</u>, the inclusion of the time dimension and, at least in prospect, the correct representation of all links and relationship (structural, causal, etc.) occurring in the product or process under analysis.

Of course at present the model is not refined, and the research far from over. There are large numbers of both critical aspects to be addressed and potentialities to be discovered and explored.

Nevertheless, the vector proposal constitutes a first step towards a fully formal, rigorous methodology, and a contribute to the construction of a complete, effective, reliable, paradigm for Functional Analysis, a task that will require the joint effort of the whole community.

The paper is organized as follows. A brief review of the critical aspects of FA is provided in section two; particular emphasis is given to those issues that point out the need to go beyond the classical approach. Section three presents the vector model and its first consequences. Section four expands the analysis of the model and comments some of its strengths and weaknesses. Finally, section five outlines the possible future directions of investigation and evolution and concludes.

2 CRITICAL ANALYSIS OF THE CLASSICAL APPROACH TO FA

2.1 Issues of internal coherence

We briefly review here the main unsatisfactory features of the classical approach to FA, leaving an in-depth analysis to a forthcoming paper.

The following points can be addresses, at least in principle, remaining within the approach itself.

They do not necessarily require a radical change in the structure of databases or in the representation of functions (although such a change can of course reveal itself a better solution than introducing a series of patches to fix the various failures).

- The criteria used to construct the Functional Basis, and in particular the synthesis represented by the RFB, are historical ones, namely the continuity with existing sets, or lexical ones, united with a minimalist attitude (leading for example to the elimination of all functions with overlaps in meaning). As a result, such sets are based on conventions and often on arbitrary choices, hence they are difficult to learn, of limited validity, and sometimes counter-intuitive.
- The small number of entries in the base, unrelated to the rest of the functional space, with no way to navigate a rich database of correspondents, means lack of precision and completeness. The representational power of the method is reduced, and high the possibility of introducing ambiguities and errors.

- The forcing used to reduce the complexity of natural language and of functional space to a small set of supposedly independent entries led in some cases not just to a limiting representation but to real errors, duplications and contradictions.

Leaving the details to a separate paper, we mention here that, in order to address such issues, we have constructed an improved Functional Basis that maintains a vertical structure similar to the current ones, but with both the vertical four levels of generality and the horizontal dimension of semantic areas based on physical, chemical and logical laws. Such construction minimizes the conventional aspects of the Basis and gets rid of inconsistencies.
Moreover, the fourth level of detailed functions is integrated in a wide functional dictionary of more than 4000 entries [9], which constitutes a powerful aid in the search for functional variants.
We intend such new Functional Basis with its tabular representation as a useful tool for practical applications that could help designers and analysts in their daily work and operative studies. However, in order to fully overcome the limits of the classical approach, a radically new, matrix-like structure based on a rigorous formal language, is necessary.

2.2 Issues concerning the inadequacy of the representation

Even if the critical aspects of the previous subsection could be fixed by carefully constructing the Functional databases or basis, there are other, deeper issues that are very difficult or impossible to address in a tree-like and natural language based structure.

1. First of all, the tree-like classification is not able to capture the complex, multi-directional nature of horizontal and vertical relationship among functions. A very important case where such father-child approach shows its limits concerns those functions that encode information about more semantic areas (a feature we have called **functional multiple inheritance,** according with the analogous phenomenon occurring in programming languages and in Linguistics) and hence should belong to more than one category. Another example regards functions that are sub-case of a more general one only under certain conditions, and so on. In a similar way, the important horizontal links of synonymy, affinity and antonymy are totally flattened or erased in a rigid hierarchical organization.
2. A partially related issue is the presence of unavoidable ambiguities and repetitions deriving from the very vagueness and complexity of the natural language. Synonyms as well as specifications or generalizations are never exact; the same verb can have more meanings (polysemy); the same function can be described by several verbs (redundancy); finally, again a function can imply other functions and refer to different semantic areas (multiple inheritance).
3. Due to the mainly taxonomical nature of the existing database, and lacking any theoretical reference frame, it is very difficult to construct grammatical rules for the combination of functions. In a similar way, it is very difficult to introduce the temporal, causal, structural relationships that may be very important in the description of real products or processes.
4. Finally, the classical approach, based on natural language and with no prescription on how to combine functions, can provide only qualitative, heuristic statements. On the other hand, when studying real products and processes, precise, quantitative comparisons would be very useful.

For the very nature of the above mentioned issues, no incremental improvement of the classical approach can solve them. We need a totally new approach, able to solve them all in one go.

3 THE VECTOR MODEL

3.1 Functions, actions, operations. Some definition.

The concept of function in literature is quite ambiguous.
In [1], a "function" was originally defined as an operation on flows, represented in the form verb+object. Shifting the perspective on the subject and on the aim of the action, TRIZ defined instead the function as the "positive meaning" (*i.e.* useful action or state) of a system to the outside world.
Later, in [2] authors defined functions as the operation only, expressed by an active verb; with the introduction of the RFB, such verbs belong to a standardized table of so called basic functions (objects as well have to be chosen from a codified taxonomy of flows) [5]. As noted by Vermaas [11] however, the product function and its sub-functions are still represented as operations on flows.

Vermaas also noted that not all the couples ‘basis verb + flows’ make sense or have sufficient representational power. In fact, a more sophisticated analysis is needed to deal with flow-restricted functions, with logical or physical boundaries, with functions where the input is different from the output and so on. Mills [12] suggested that context information (i.e. physical, chemical, mechanical laws) select the admissible verb-flow couples.

Without entering in an in-depth discussion about the precise definition of the concept of function and all the related consequences, here we would like to enunciate a definition that is consistent with the vector representation we are constructing:

“*A function is an action performed on a flow, that evolves in time a given aspect (preferably measurable) of the flow*.”

Some notes about the above definition and its link with vector representations:

- The entity modifying the flow is called ‘action’; we will use the term ‘operation’ in its mathematical meaning, to indicate operation between vectors (in mathematics and logic an operation is a procedure which produces a new element from one or more input elements).
- Flows may be both subject and object of the action; the action is aimed at the object but as a result of the mutual interaction the state of the subject may be modified as well.
- The definition introduces the novel notion (derived from physics) of time evolution in both its meanings of changing and of preventing from changing. The inclusion of the time dimension overcomes the standard static representation of a system as a series of consecutive pictures.
- The new description is not based on (too) generic flows but it recovers the main physical information about the particular aspects of the flows actually modified by the action.
  Indeed, only few very abstract functions really carry an action on the flow as a whole; rather, they evolve only an aspect of it. It is probably needless to underline the total absence of information in the function "convert material (into material)" - and the need for a specification of which property of the material has really been changed by the conversion function.
- Last but not least, shifting the focus on the physics of the system, the new definition does not constrain the representation to the limiting verb+object form. In fact, the idea of an action that merely combines with a flow to give a function is too simplistic, actually the way action and flow are correlated implies more information (concerning the time evolution, the modality of interaction, the speed of the phenomenon, etc...) than those usually considered in standard FA.
  As an example, consider again the action *convert*. If the old picture verb+object were correct, the couple convert+liquid would already be a good function, while it is clearly not suitable to discriminate between *evaporate* and *solidify*. In fact, already standard FA contradicts the verb+object form by adding the output specification. However, classical FA still loses the relevant information about the different entropic change between, say, *boil* and *evaporate*.
  Our definition allows instead to include all necessary information and relation in the function.

The last example, together with the introduction of time, indicates that actions and flows are now just formal elements, defined to carry quanta of information; in order to fully define the space of functions, further, non-verbal, directions of information content can be introduced. Overall functions are the result of a particular operation, in mathematical sense, between actions, flows and, when necessary, other formal elements so that the resulting object encodes all relevant physical or logical information.

We are now ready to construct the vector representation of functions.

3.2 The Function Vector Space

In mathematics, a vector space **V** (on the field of real numbers $\Re$) is defined as a space closed with respect to two operations:

a) multiplication by a real number: if **v** in **V** then λ **v** in **V** too (with λ in $\Re$).
b) addition between elements of the space: if **v**, **w** in V then **z=v+w** in **V**.

A basis is a set of **linearly independent** elements that span the **whole** space.

The space of functions already exists, at least as a taxonomy. In order to introduce a vector structure into it is hence necessary to define the two operations of addition and multiplication by real number, and demonstrate that the space is closed with respect to them.

For what concerns **multiplication**, the starting point is the realization that both hierarchical and synonymy relationships can be represented as directions in the functional space. We can define the multiplying real numbers as the relative weight between two actions belonging to the same direction.

As an example, consider the verbs *walk* and *run*: they can be seen as two different linear combinations (by a suitable λ factor) of the same functional concept. In this case the common concept is the *motion by step* and the real number λ gives the intensity of the two motion regimes (i.e., if we identify the *motion by step* as a unit vector **m**, *walk* and *run* will be $\lambda_1\mathbf{m}$ and $\lambda_2\mathbf{m}$**).**

Of course such intensity can be measured as a speed (m/sec), frequency (step/minute) or energetic power consumption (hearth rate or Cal/hour). The choice depends on the parameters we are interested in. However it is possible to pass from one to the other by conversions as it happens in standard vector space when performing a change of measuring units. Clearly the relationship between two different units can also be non-linear (think at the conversion from step/minute to Cal/hour in the example at hand). It is finally worth noting that it is not important to define λ_1 and λ_2 but the ratio between them because **m** is a formal, abstract element while $\lambda_1\mathbf{m}$ and $\lambda_2\mathbf{m}$ represent the real actions.

Horizontal and vertical relationships constitute the key for defining the **addition** operation as well.

As synonyms share a common direction, many actions belong to two or more functional directions/concepts. The more evident example is that of multiple inheritance (that implies also hierarchical dependency) described in section 2.

For example, the action corresponding to *conveying* implies both the general semantic category of *movement*, the transfer of something from a place to another, and the category of *binding* the flow to a certain path by limiting its motion in the unwanted directions (with pipes, banks, etc.).

Multiple inheritance and more generally the simultaneous presence of more than one semantic concept or physical effect is hence the mathematical counterpart of adding two vectors to generate a new one.

Therefore, continuing with the example, if we identify the action *to convey* with the vector **c**, the category of *movement* with the unit vector **v** and that of *binding* with **u**, then $\mathbf{c}=\mathbf{v}+\mathbf{u}$.

Of course, a function can have different degrees of dependency from each of its generating categories. The multiplication operation just defined allows to discriminate by weighting the contributions.

For example, the actions *deviate* and *bounce* have the same fathers (*obstacle* and *repulse*), but deviate is more a category of obstacle with repulse as a secondary information, while for bounce is the other way round. If we identify *obstacle* and *repulse* with the vectors **o** and **r**, and *deviate* and *bounce* with **d** and **b** it is possible to write $\mathbf{d}=\lambda_3\mathbf{o}+\lambda_4\mathbf{r}$ where $\lambda_3>\lambda_4$ and $\mathbf{b}=\lambda_5\mathbf{o}+\lambda_6\mathbf{r}$ where $\lambda_5<\lambda_6$.

It is clear that all known functions can be represented in this way. In the worse case, they just define a new direction on their own. On the other hand, closeness also requires that all linear combinations of functions are still in the space. From a formal point of view this is certainly true. Whether the generated elements are new true functions or not will be discussed later.

Let just say here that real functions are subject to physical laws; such laws can be translated into algebraic constraints determining hyper-surfaces within the functional space. True functions will be bound to live on those hyper-surfaces.

3.3 A vector basis for the functional space

The space is now defined. To manage it and better exploit its potential, it is very useful to introduce a basis, in analogy with what happens in standard linear algebra.

Therefore it is necessary and convenient to determine a complete (and smartly chosen) set of basic elements. Every action will be described uniquely by a linear combination (in the true mathematical meaning) of these basis' elements. This way the translation to a formal language becomes total.

The notion of basis is a major step toward the implementation of a fully formal grammar and of course the existence of a functional basis is not a new idea. Actually already Little [10] and Hirtz [5] assume the existence of a functional space and define a "basis" for that space. Their set has however a pragmatic aim and the reference to mathematical basis is just an analogy. No operations or other formal structure that can guarantee linear independence and completeness are defined

The idea here is to extract *n* basic concepts (not actions) e_n, not overlapping and derived from physics (kinematics, dynamics, etc..) or from logics (union, intersection, addition, subtraction). Such a set will constitute our basis, while physical and/or logical laws also supply the rules to combine them. Linear algebra prescribes the tools to operate with the elements. We remark that the basic concepts e_n are formal elements neither actions nor functions nor words. It is their combination that allows to build/describe actions and functions.

3.4 Guidelines for constructing a basis

The general requirement that must be fulfilled is of course that of linear independence and completeness of the chosen set. Such check allows selecting the entries consistently.
To determine the actual elements of the basis, the following criteria can be followed:

- Semantic hierarchy;
- Correspondence with physical effects or logical concepts;
- Synonymy and antonymy.

The first criterion is clearly the closest to the historical procedure used in [4], [5] or analogous works.
Of course the Functional Basis of the standard approach already individuate classes of generality that, after careful checks, can be transferred to the new picture. Two procedures are possible: a top-down and a bottom up one. The top down one could be similar to the one we have followed:

1. all the verbs belonging to the first three levels of the RFB have been collected in an unique class; some verbs belonging to TRIZ [6] or Hundal's [12] work have been reintroduced,
2. then a series of "lines of action" (each with two directions corresponding to each couple of action/anti-action) have been selected,
3. the concept shared by the two elements of each line of action has been considered as basis vector.

The bottom-up approach starts from the extended databases (such as that of [9]) and distillate the common concepts, still using linear independence as a guide. Multiple inheritance is also a useful guide: functions with common fathers will live in the intersection of two hyperplanes and the vector generating such direction is clearly a candidate basis element.
A similar bottom-up approach can also be used for the third criterion, synonymy and antonymy.
After collecting all synonyms for each area, one individuates the common concept and promotes it as a basis vector. In the same way, if two functions are antonyms, there must exist a concept that they share with opposite implications.
After the semantic screening on existing databases has been performed, it is necessary to introduce physical and logical considerations. On one hand the consistency with physical laws allow to check the semantic construction, and guarantee linear independence avoiding overlapping concepts. On the other hand it is possible to introduce all those sectors or concepts that a purely linguistic analysis is not able to grasp, such as physical effects (piezoelectric effect, photovoltaic effect and so on).

3.4.1 An example of a core/nucleus for a new vector basis

Even if the aim of this paper is to define the baseline for the introduction of a formal approach in functional analysis, with no claim of exhaustiveness, we provide here an example of reorganization of the standard functions appearing in the literature in a way compatible with a vector interpretation.
It is very important to note that the following table is not our proposal for the ultimate vector basis.
It is just a reorganization and integration of some of the existing entries of previous functional basis or databases (mainly from RFB [5]) according to a different philosophy, in order to illustrate the main differences of the new approach with respect to the classical ones.
The entries have been paired in couples of function/anti-function (last two columns). Each pair singles out a direction or an hyperplane of meaning that is independent from the others (an hyperplane may be decomposed using more detailed functions). Some entries are actually obtained combining two of such directions (e.g. *to guide*). Finally, semantically related directions (sharing the same logical or physical context of reference) can be considered close in the vector space and hence belonging to homogeneous subspaces (first column). We note here that the structure presented in table 1 is not totally new. Classification according to physical and logical inputs (but without a vector interpretation) has been done extensively in the first phase of Functional Analysis development (see for example the works of [2], [19], [20], [21]). While following research concentrated on small sets of natural language entries, recovering inputs from such early works can be a valid aid in constructing a rigorous vector basis.

3.5 More on the relationship between functions

Once the vector space and a base are defined, linear algebra provides the tool to represent functional relationships in a rigorous way.
The notion of scalar product for example, providing the angle between two directions in the functional space, allows determining the degree of affinity between functions in a quantitative and rigorous way.

Two actions with no relationship whatsoever will be represented by orthogonal vectors, *i.e.,* their scalar product would be zero.
A non zero scalar product of a vector with a basis vector implies a hierarchical, father/son relationship, while a not-null scalar product between two actions indicates a synonymy relationship.
To be more precise, in hierarchical relationship the conceptual category to which sub-functions belong is given by the hyperplane spanned by all subfunctions; the generating father is individuated by the common intersection of all functional directions of such hyperplane.
Another useful notion is that of projection.
Apart from a few very general ones, all functions refer to (a class of) specific objects only: one can **record** only signals, or **to percolate** can only refer to a liquid. These functions have been called by Vermaas “flow restricted” functions. It means that objects of functions are subject to extremely precise definition in physical terms. Such feature is represented in the functional space by suitable projection operator between the sub-space of actions and that of flows. Each projector encodes the information about the allowed physic interactions.
Physical constraints also define the hyper-surfaces in the combined vector space (actions, plus flows plus other non verbal information elements) were real functions can live.
Different hyper-surfaces individuate areas of physical or logical meaning, and by extension may determine the application domain of the functions living on them.

Table 1. Extract of a functional basis compatible with a vector structure

Area	i	Formal element Direction of the action/anti-action	X_i	$\overline{X}_i$
Processes related to the increment of a certain parameter	1	Creation	Create	Destroy
	2	Addition	Add	Remove
	3	Qualitative change	Improve	Degrade
	4	Quantitative change	Increase	Decrease
Processes of Junction	5	Reversible junction	Assemble	Disassemble
	6	Irreversible junction	Join	Separate
Processes related to Information	7	Gathering of information	Detect	--
	8	Transfer of Information to the user	Display	Hide
	9	Information filtering	Select	--
Processes related to Kinematics principles	10	Enable dof	Allow	Obstacle
	11	Change position	Move	Arrest
	12	Change position + Constrain dof	Guide	--
	13	Constrain dof	Release	Block
Processes of Exchange	14	Exchange with environment	Import	Export
	15	Provision	Store	Supply
Processes related to Functioning	16	Functioning at t=0	Start	Stop
	17	Functioning at t≠0	Operate	Pause
..	..	…	…	…

Standard Base Elements New additions Assimilable (to Hirtz’s) -- No anti-action

4 FURTHER COMMENTS

4.1 The importance of correspondents

Correspondents of RFB are not a fourth level, but just a list of duplicates of the third level base verb and are usually neglected in the analysis. Actually in classical approaches there are only few macro-concepts that constitute the base and all the remaining functions are dismissed as redundant, being considered identical or subcategories of the basic ones.
Conversely, the importance of the correspondents in the presented approach is greater than in the standard one. Here they become sources for functional alternatives/variants and bring important information for completing the vector framework.

4.2 More on physical hypersurfaces and real functions

As we have seen, using the physical laws it is possible to identify some logically related areas (namely hyperplanes spanned by a subset of vectors) that we call Areas. In an intuitive picture they can be thought as hypersurfaces permitted by the same class of physics (statics, dynamics, kinematics etc.). Therefore we use the areas (space-kinematics, energetic,. thermodynamic, logics, information management, etc) just as an aid for the reader.

Clearly, there are some function not yet classified because of lacking the verb (i.e. some physical effects as piezoelectric effect do not have a corresponding verb) or because they are combination of actions, structural (i.e.: constitute, arrange, surmount, confine) or logical information (i.e. wait, imply) or finally because not yet discovered. Moreover the existence of some areas is determined also by the flows on which the action interacts. Hence it emerges the need to expand the analysis to other functional elements such as flows, physical, structural and logical parameters.

A key note is that not all the possible linear combinations of actions (actions and flows, and in future parameters) can have a corresponding known or admitted action.

More generally, the full vector space, spanned by all possible linear combination of all basis vectors, is much larger than the set of verbs of the English language. Just to mention an evident reason for that, there is no elementary function (that is not taking into account composite functions or series of temporally/causally consequent steps) that can be spanned by all base vectors at the same time or by two basis vectors which refers to incompatible modalities. Known functions will distribute along intersecting hypersurfaces, spanned by the single elements. This does not mean that the other points of the vector space do not have any interest, let alone the internal consistency of the space. Indeed, multi-base functions for example can be related, once the temporal dimension is taken into account, to composite functions. Moreover many linear combinations allow to describe a whole set of new functions that are not defined with a verb in the English language, but are nevertheless implied by the use of particular materials or the presence of certain physical effects. To sum up, real actions will live in hypersurfaces admitted by physics, other points admitted as linear combinations of base vectors can generate non-real actions. That does not diminish the value of the approach.

Actually not even all real formal functions are useful in practical applications. For example anti-join is better labelled with the verb separate. In a similar way the opposite of detect, to *conceal* (information) is seldom used in technical descriptions. Actually *conceal* has few sub-cases against the large numbers of kind of detections. One may wonder whether this lessen the validity of the concept of opposite function. The solution is that the great number of functional variants of detect do not come from the action part/ basis vector, but from the additional information (co-fathers, modalities) and from the different objects one could detect. An anti-action is NOT the anti-function.

Again it would be the physics of real phenomena to tell us which directions of the functional space are not just existing, but also populated.

4.3 More on the relation between physics, engineering design and functions

At this point it may be useful to remark that we are not claiming that functions can be derived only from physical principles (and chemical and logical ones). An artefact exists according to technical specifications and to the law of physics but it is there to fulfil human needs and values. Functions are exactly the connection between the two point of views, capturing the essence of what the product does and why. The dichotomy between goal (from the user point of view) and physical implementation is typical of all FA and is often reflected in the overall function vs sub-functions decomposition [12].

A good representation of functions should be unique, faithful, unambiguous, and capture both aspects. The first idea is that physical inputs are a guide for rigorous and effective representations. Functions should always be expressed in such a way to make explicit the physical effect implied. We should always ask "which state of an observable variable is affected by this function"? It is clear that most functions in everyday language are not formulated this way. But this remark applies also to part of the literature, in which functions are expressed as "verb + object", but often the physical content of the verbal or object element is hidden or not at all clear.

Of course the above statement does not mean that the description used for functions must be identical to that used for physical laws (such as equations and so on). Actually the emphasis given on the concept of action in the definition of function and the complex nature of products and processes that Functional Analysis deal with imposes that in most cases the two descriptions will be formulated in

substantially different ways. The sense of the statement is that a parallel between the two alternative descriptions must always be possible.
Our second claim is that with the vector approach it is possible to represent both the physical part of the action *and* its goal (we note that the notion of action implies an intention by the agent).
The two aspects are just different components, belonging to different subspaces, generated by different sets of basis elements. The function useful in engineering design is obtained by summing the two contributions.

4.4 Changing Basis

A basis of a true vector space can be rather arbitrary, since the transformation from a basis to another is always well defined. Our base is indeed so constructed. Hence, the conventionality implicit in any choice of which verbs of the natural language are promoted to representatives of abstract actions is no longer a problem: it is well known (and easy) how to shift from any description to any another.
On the contrary, in the standard taxonomical approaches, every choice for which verb is a base one and which is a mere correspondent is qualitatively different: it changes in a non foreseeable manner and hence the functional descriptions of artefacts that derives from it is largely arbitrary.
Moreover, in particular with the introduction of temporal/logical/structural dimensions via the specifiers/parameters, new verbs fall almost automatically in their right place in the space.

4.5 Action/Anti-action: a resource and a possible source of ambiguities

Another novelty introduced here is the idea that for each action A exists an opposite action (or anti-action) $\overline{A}$ such as the sequence of A and $\overline{A}$ on an object does not alter the characteristics of the object itself or equivalently it corresponds to no action at all. Mathematically $A \ \overline{A}$ is equivalent to **1** (identity operator). There are several possibilities for the actual representation of the opposite function. Mathematically, the key point is the operation to which the element $\overline{A}$ refers to. If it is addition then it will be -A, if multiplication 1/A, if logical connection NOT A, and so on.
In the same way, we should provide several examples of couples action/anti-action where different notations could fit better with their meaning in natural language. Many of the apparent logical problems are just a misleading consequence of the standard lexical approach. The notation proposed here has the aim to generalise all these cases. Speaking of formal objects there is no problem of meaning and of complying with the meaning of the actions.
The formal construction just automatically takes into account all possibilities. Moreover, a vector will have many components and lexical contraries or antonyms may correspond to different combination of inverted components. The already mentioned example of *detect* and *conceal* clearly explains such statement.

4.6 Time dimension

The time evolution can be implemented adding a temporal dimension in the vector description of the device. All descriptive issues /functions are taken to be simultaneous, that is frozen at t=0, but some sub-function can extend its vector along the new dimension, spanned by the base vector e_t
Functions with a temporal evolution have a slope in this direction, and the projections on t=0 and t=1 describe the initial and final state of the process. The ordinary FA space is like photography, a subset of the whole projection of the full space on the a-temporal, a-spatial hyperplane.
Such additional direction may prove a very useful tool, but we have not tested its use so far though, it is not totally clear what its actual impact on the analysis could be.

4.7 Strengths and Weakness of the model.

The elaboration of a full vector approach is in progress and of course many aspects are still to be cleared. In our opinion however some strength of the model, also with respect to standard taxonomical approaches are already manifest.

1. The novel vector approach is scalable, covers a wide range of actions even maintaining a low number of basic elements (we aim at a classification that takes inspiration from elementary particle physics, where the orientation of the same restricted number of basic vector quantities allow to classify without ambiguities a huge number of derived objects).

2. The wide Functional Database associated to the approach covers all natural language terms that the designer may find or want to use, allowing him to perform the analysis in way closer to human natural intuition. The Database automatically gives information about all functional relationships and translate the specific functions in term of the Functional Basis, thus constituting a valid aid to the overall usability of FA.
3. It is easy to formally describe the semantic coexistence of two categories of actions in the meaning of a verb: an action is just the linear combination of two base vectors. The introduction of formal elements solves automatically the ambiguities due the use of a natural language translating each action in an unambiguous mathematical formulation.
4. Similarity relationships can be dealt with in a very precise way: the number of shared basic elements together with the possible connection with the flows' space determines the degree of similarity. Opposition relationship, too, has a simple interpretation, with a rich (yet potentially ambiguous) range of possible declinations that can represent all possible situations.
5. The chosen architecture shows a deeper attention to the physic of the system, forces the users toward objectiveness and allows them to perform measurements. It is a step for moving FA towards a more scientific approach.
6. It allows to describe a whole set of new actions and functions that are not defined with a verb in the English language, but are nevertheless implied by the use of particular materials (e.g. piezoelectric) or the presence of certain physical effects (skin effect, tunnel effect, etc.).
7. A vector approach has an enormous advantage in systematic search: it is possible to take into account all variants and synonyms in an arbitrarily large neighbourhood of the starting vector, drawing from the 4000 entries database [9], without loosing the information that vectors gather around basic directions, that is the abstraction categories.
8. It is easy to disambiguate both hierarchy and horizontal structure: both ordering are dictated by the sharing of basic vectors and by the allowed connection with flows. As a consequence, the derived Functional Database is more coherent and objective than the previous ones.
   It allows the ordering of categories and classification in a effective and rigorous way, eliminating or drastically reducing ambiguities.
9. Last but by no mean least, the vector approach naturally allows to make quantitative evaluations. The degree of similarity between two functions can be expressed by scalar products and subsequently used to elaborate algorithms to compare overall products.

As a further example on the novelties introduced by the vector approach, we have performed a series of case studies to show the advantages with respect to standard functional basis.
In the first test, we have asked 20 engineers, all knowing the principles of FA, to distribute a series of electromechanical functions (Rotate, Weld, Translate, Glue, Solidify, Rivet, Screw, Liquefy and Vibrate, etc..) among the three RFB classes of Move, Join and Transform.
They had to rely on their intuition only: they were not provided any clue on how to do it, neither a basis or list of correspondents/synonyms.
While certain functions were clearly assigned to one class (Rotate under Move in figure 1a), many functions were positioned half way between two classes (as Weld in the figure 1a). It is a clear evidence of multiple inheritance that classical approach fails to capture.
The degree of mixing between the two fathers in case of multiple inheritance also varied, as shown in figure 1b for two functions sharing the same two semantic classes. Glue is associated to the class Join to a larger degree than welding which has in turn a bigger component along the transform direction.
The statistical data can be used to determine a quantitative measure. In vector terms we can say that, being **J** and **T** the basis vector associated to the basic actions of joining and transforming, the vector *glue* is about 0.85 **J** +0.15 **T**, while *weld* is about 0.6 **J** +0.4 **T**. The same can be done for *rivet* and *screw*. That result in accordance with the theoretical intuition. The scalar product between the two vectors give a quantitative indication of the degree of similarity between glue and weld.
It goes without saying that such analysis is impossible using the classical approach.
In a second test, we have asked other 20 engineers to construct a hierarchical graph connecting a series of functions, all belonging to the RFB (among them convey, channel, distribute, allow DOF, constrain, guide, separate, move, transfer). The result about which functions were more general varied considerably. Notably channel, that in RFB is more general than all others, has always been put under either guide or constrain or transfer, with almost the same percentage for the three entries (figure 1d).

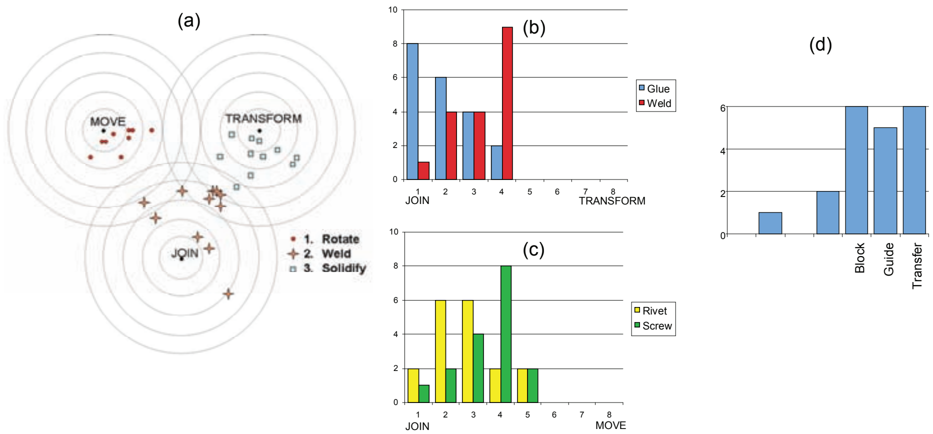

Figure 1. a)Radar scheme; b)Glue vs Weld; c) Rivet vs Screw

Of course some critics and unresolved points remain. Our belief is that many of them will be overcome by the evolution of the model in an even more sophisticated formal language.

1. There still is some residual overlapping in the meaning of those representatives that have been chosen from verbs of natural language. In future developments, there is the clear need for strict definitions (e.g. Import vs Supply, in particular for mechanical parts).
2. Some functions are so specific they almost constitute a class on their own. The danger is the unwanted proliferation of directions in the space.
3. Each choice of the coefficients may vary if the boundary conditions of the problem change.
4. The focus of the analysis must be clearly stated: *e.g.* all signal related functions have different decomposition when considering the point of view of the observer or that of the observed system. See Vermaas [11] for a wider discussion on this topic.

5 CONCLUSIONS AND FUTURE DEVELOPMENTS

The vector approach to Functional Analysis we have introduced is an interesting and potentially extremely useful methodology, let alone an interesting theoretical frame in itself.

In our opinion its main importance is that introduces in the field a new direction of studies, perhaps a revolutionary one: the use of formal languages and mathematically inspired models to represent the functional world.

Of course the vector approach alone is just a first step, but it is not sufficient.

The current direction of our research is the extension of the vector approach from the sub-space of actions to the whole space of functions (that is the tensor product of sub-spaces of actions and flows[11]) and afterwards to the alternative space of physical functions (actions plus flows plus parameters, introducing elements from causal, temporal, structural dimensions [12]).

An example will help to understand the complexity of the formal framework at the horizon.

Let us consider the three concepts of *evaporate*, *condensate* and *boil*.

To represent *evaporate* we can concatenate in a unique vector (using the tensor product of linear algebra) the single vectors coming from actions' and flows' sub-spaces according to the path (action | initial state flow | final state flow) thus obtaining transform|liquid|vapor. In the same way *condensate* becomes transform|vapor|liquid. However, if want to distinguish between the functions *to boil* and *to evaporate,* that would not be enough and we should introduce further elements, belonging for the example to the vector subspace of parameters. We can therefore imagine that *evaporate* will specify to the sequence transform|liquid|vapour|reversibile while *boil* to transform|liquid|vapour|irreversibile.

All the above considerations concern the single functions alone.

The subsequent, hard task would be the translation in a mathematical language of all structural, causal, logical relationship that can occur between functions. If the formal representation of a single function as a vector seems to be accomplished, we still do not know how to treat and represent structured lists of vectors/functions. Of course the structured lists we have in mind are the overall functional

descriptions of technical products and processes. The implications of finding such formal representation are obvious.
A possible aid may come from the concept of time evolution of the physical characteristics of the system. Investigations in this direction are currently in progress
In any case, the vector approach may constitute the starting point also for the ambitious goal of representing in a formal way the complete product and not just its constituent functions, a task that will require the joint effort of the whole community.

AKNOWLEDGEMENTS

The authors would like to thank Ms. Donata Gabelloni and Mr. Dario Tambellini for their thesis works on both technical and methodological aspects. The authors would also like to thank Dr. Stefano Borgo and Dr. Pieter Vermaas for very useful discussions. The financial support of PRIME Network of Excellences (FP6th), Italian Ministry of Research FIRB-IRIS and PRIN2006 is kindly acknowledged.

REFERENCES

[1] Miles L.D. Technique of Value Analysis and Engineering, 1972 (McGraw-Hill Book Company)
[2] Pahl G. and Beitz W. Engineering Design: A Systematic Approach, 1984 (Design Council, London).
[3] Szykman S., Racz J. W, Sriram R.D The Representation of Function in Computer-based Design, ASME Design Engineering Technical Conferences, DETC99/DTM-8742, 1999, Las Vegas, NV
[4] Stone R. and Wood K. Development of a Functional Basis for Design, Journal of Mechanical Design, 2000, 122(4), 359-370.
[5] Hirtz J., Stone R., McAdams D., Szykman S. Wood K., A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts, *Research in Eng. Des.*, 2002, 13, 65-82.
[6] Altshuller, G., “Creativity As An Exact Science”, (New York, Gordon And Breach, 1988).
[7] Nagel, R. L., Vucovich, J. P., Stone, R. B. and McAdams, D. A. “Signal Flow Grammar, From the Functional Basis”, *International Conference on Engineering Design*, ICED’07 Paris, France
[8] Sridharan, P. and Campbell, M.l. (2004) A Grammar for Functional Structures. Design *Engineering Technical Conferences and Computers and Information in Engineering Conference* (ASME, Salt Lake City, Utah, 2004).
[9] Bonaccorsi, A., Fantoni, G., 2007, “Expanding the functional ontology in conceptual design, International Conference on Engineering Design, Iced’07, Paris, France.
[10] Little, A., Wood, K., McAdams, D. (1997), ”Functional Analysis: A Fundamental Empirical Study for Reverse Engineering, Benchmarking and Redesign”, Proceedings of the ASME Design Theory and Methodology Conference, Sacramento, California
[11] Vermaas P. "The Functional Modelling Account of Stone and Wood: Some Critical Remarks", International Conference on Engineering Design, ICED’07 Paris, France
[12] Mills J. J. and Iyer, G., Using Domains to Constrain Design Variables, International Conference on Engineering Design, ICED’07 Paris, France
[13] Hundal, M. (1990) "A Systematic Method for Developing Function Structures, Solutions and Concept Variants", Mech. Mach. Theory, 25(3):243-256
[14] Collins, J., Hagan, B., and Bratt, H. (1976), ”The Failure-Experience Matrix - a Useful Design Tool”, Transactions of the ASME, Series B, Journal of Engineering in Industry, 98: 1074-1079
[15] Rodenacker, W.G., “Methodiches Konstruieren”, Springer Verlag, Berlin, 1976.
[16] Roth, K., “Konstruieren mit Konstruktion Katalogen”, Springer Verlag, Berlin, 1982.
[17] Kitamura Y. Kasai T. Mizoguchi R. Ontology-based description of functional design knowledge and its use in a functional way server, 2001, Pacific Asian Conference on Intelligent Systems.

Contact: Gualtiero Fantoni
Department of Mechanical, Nuclear and Production Engineering, University of Pisa
Via Bonanno Pisano, 25b, I-56126 PISA, Italy
Phone +39 050 913028
e-mail g.fantoni@ing.unipi.it