Amaresh Chakrabarti
Associate Professor
Centre for Product Design and Manufacturing
Indian Institute of Science, Bangalore 560012,
India
e-mail: ac123@cpdm.iisc.ernet.in

A New Approach to Structure Sharing

Structure sharing means fulfilment of several functions or functional properties by the same physical structure, and is an important concept in product design. However, few guidelines and methods for supporting structure sharing, especially on computers, are currently available. The aim of this paper is to present a new, computational approach, for supporting structure sharing in design, that can automatically create, and offer designers for evaluation, a variety of alternative solution principles as well as their potential, minimal, qualitative embodiments that can fulfil a given intended sensor functionality. These potential alternatives are structure-shared where possible.

[DOI: 10.1115/1.1683817]

1 Introduction to the Problem

Structure sharing means fulfilment of several functions or functional properties by the same physical structure. The concept was popularized in [1], using the term 'function sharing' to describe it. Structure sharing is an important concept in product design, and is often referred to using terms such as combination of functions [2] or integrated structures [3]. These concepts have been used consciously or unconsciously in making products more innovative and efficient. However, while the importance of these concepts has often been emphasised in literature, principles embodying these and approaches and methods for supporting these, especially on computers, have rarely been investigated in any depth.

Structure sharing is one of the four categories of sharing [4], the other three being *function sharing* (sharing of the same function by several structures), *structural redundancy* (providing the same function by co-existing alternative structures), and *multi-mode integration* (the same structure providing different alternative functions). While structure sharing has the positive benefit of decreasing the use of resources (e.g., size, volume, weight, overall cost, etc) in making a product, it can also have the negative impact of decreasing its changeability (e.g., ease of adjustability, disassembly, repair, and reuse of parts). However, there are many areas, such as aerospace applications, where minimum use of resources is of prime concern, and structure sharing has widespread use.

This paper presents a new approach for supporting structure sharing in design that has been implemented into a software for automatically creating, and offering designers for evaluation, a variety of alternative solution principles as well as potential, minimal, qualitative embodiments (termed here as conceptual structures) of sensors that can fulfil a given intended sensor functionality. These potential alternatives are structure-shared where possible. For instance, given the functionality of sensing a force with a voltage, the software suggests a variety of alternative principles including that of using a surface area to develop a stress from the force, a piezo-electric effect for developing charge from that stress, and a capacitance for developing a voltage from the charge. It then generates many alternative conceptual structures for each of these principles, e.g., one that uses a piezo-crystal, having an *area* as well as *piezo* and *dielectric* properties to respectively activate force-stress, piezo-electric and capacitance effects, all within the same component.

The work reported is a subtask within a larger framework being developed for supporting designers to explore the widest selection of solution principles—a principal task in conceptual design—for subsequent embodiment into viable concept variants; need for such a framework, especially for sensor design has been articulated in [5]. In three previous papers [6–8] we described the framework which is intended to support alternative formulations of device functionality, generate and offer a wide range of alternative solution principles to fulfil the intended functionality, and help designers embody and envision these principles. A building blocks approach to automated synthesis of solution principles was reported in [6]. This paper reports how these principles are automatically transformed into potential, structure-shared conceptual structures. The rest of the paper reviews related work, the approach, its implementation and evaluation.

2 Previous Work

The broad context within which the present work is embedded requires that the widest possible range of solution principles are generated to solve a design problem (we call this the synthesis of solution principles), and these principles are embodied to the extent necessary for their effects to be activated (we call these minimal embodiments conceptual structures, and their generation synthesis of conceptual structures) with structure sharing where possible. These conceptual structures can then be evaluated using various methods including identification of potential side effects to which they are susceptible, see in [7]. This section is divided into two sub-sections: one focusing on the synthesis of solution principles and conceptual structures, and the other on structure sharing of these.

2.1 Literature on Synthesis of Solution Principles. Several researchers worked on synthesis of solution principles [1,9–14]. Some worked on automated synthesis of a single solution principle [12,13] or multiple, alternative principles [1,9], while others on synthesis support [10,11,14]. There are several approaches to synthesis: design from first principles [15], systematic design using design catalogues [16], using compositional synthesis [most of the above], and using design grammars [17]. Few of these aid development of any form of embodiment for these principles, with the exception of [1,14], which are similar in that they both use bond graphs to represent solutions at the principle level, and replace bond graph chunks of a principle by components at the embodiment level. While [1] applies to single input output systems only, [14] extends this further to multiple input output systems. Our work is based on compositional synthesis, for developing both solution principles and conceptual structures. This is because compositional synthesis has a higher potential of generating innovative designs, although with high potential risks (as with innovative designs in general) and a more resource-intensive development process than the safer case based systems [18,19].

2.2 Literature on Structure Sharing. Many researchers have emphasised the importance of structure sharing. Some iden-

---

Contributed by the Computer Aided Product Development (CAPD) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received May 2003; Revised January 2004. Guest Editors: I. Horvath and D. Rosen.

tified the levels of product abstraction at which structure sharing takes place, e.g., in [20,21] that a component in a product often embodies several 'organ structures'. However, no method is developed to translate 'organs' into multi-organ component structures. Several researchers emphasised the importance of product properties as the link between organs and components, with [22] giving some empirical underpinning to this idea.

An interesting insight in this connection is given in [13], where the claim is that for a principle, constituted as a chain of physical effects, the potential number of distinct potential combinations of 'schematics' (that embody each effect) with or without 'functional overlappings' is $2^{(n-1)}$, where n is the total number of physical effects used in the principle. This provides insight into where (i.e., among which effects) structure sharing could take place. However, the possible number of schematic combinations (i.e., conceptual structures in our case) is actually much larger, and depends on which properties and constraints must be present for activation of an effect, and to what extent the chosen schematics (or conceptual structures) are able to provide these. In other words, functional overlapping between two effects is not a binary issue (no or full overlapping), but rather a continuum issue, since the properties necessary for activation of two effects may be shared by two schematics, completely provided by a single schematic, or those required for each effect provided exclusively by one schematic each.

The main work available on computational structure sharing is [1] where an embodiment developed is further function-shared by deleting some of the components within the embodiment and checking to see if the additional properties of the rest of the components can still fulfil the functions of the deleted structures. If this is possible, the component is definitely removed from the embodiment; otherwise it is reinserted. The program performs this test for all the components in an embodiment, and the outcome is an embodiment that possesses improved structure sharing.

There are three potential problems of using this approach for our project, which aims to develop, and offer designers for exploration a variety of alternative conceptual structures, structure-shared if possible, that are essential for a solution principle to work. The first is that the approach operates at a geometric level while we need to operate at a more conceptual level. The second is that it encourages structure sharing along the lines of adaptive variation rather than generative variation [23]. This makes it difficult to generate a range of alternative embodiments and allow choice between resource-effectiveness and changeability [4]. The third problem is the relative lack of computational efficiency and effectiveness of its generate-delete-test-(reinstate) loop, which would miss mutually dependent sharing options and need relatively more resources to run. The proposed approach is intended to alleviate these problems. For an analytical comparison of resource-efficiency of this approach with the approach proposed, see Appendix 1.

3 Objectives and Research Method

There are three main objectives: for a given intended sensor function, first develop solution principles, then develop conceptual structures for each solution principle, and then, enable structure sharing among components of the conceptual structures if possible.

The research method has five steps: (1) data collection and analysis, (2) development of representation, (3) development of reasoning procedure, (4) implementation, and, (5) testing and evaluation. These are discussed in the following sections.

4 Data Collection and Analysis

Textual data on eleven cases (i.e., family of sensors) was collected from several books and catalogues [e.g., [24–26]] into a single document for each case, which was then analyzed. Data analysis consisted of identifying how each sensor works at the principle level (i.e., what effects are activated and how they fulfil the overall function) and at the embodiment level (i.e., what components are essential and how they fulfil the overall function for a given principle). This revealed, among others, that some preconditions must always be satisfied for an effect to be activated. The preconditions are existence of external variables, as well as characteristics of the components or interfaces of, or constraints on the structure of the sensor. These structures are described only by components having the attributes that they must possess in order to be able to activate the effects in the solution principle they represent, thus they are called *conceptual structures*.

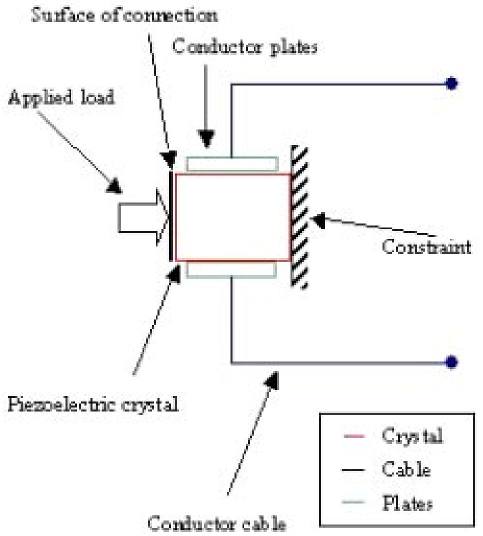

Fig. 1 **Conceptual structure of a piezo-electric sensor**

For instance, in order to work, a piezo-electric force sensor needs the force input on one of its surfaces. This input is converted into a stress by its surface area if the crystal is constrained against movement in the direction of the applied force. As a result, the crystal lattice is deformed, leading the piezo-crystal properties to generate a charge inside the material because of Piezo-effect. With two conductor plates placed on opposite surfaces of the crystal, the material shows capacitance behavior due to its dielectric properties. A potential difference between the two plates is created due to Capacitance effect, which can be sensed by a voltmeter as a measure of the applied force (see Fig. 1).

5 Representation

Our goal is to generate solution principles to fulfil a given device functionality, and conceptual structures (that provide the components, interfaces and constraints essential for the effects in a solution principle to work), to a given solution principle. Therefore, we need to represent device functionality, solution principles, and conceptual structures for a given solution principle, and identify how these are linked. Five constructs are developed to represent these: *variables, properties, constraints, effects* and *components*.

A *variable* is a quantity, associated with the system, that can vary as a result of activation of physical effects operating within the system. Most variables are input or output of a physical effect. For instance, the input and output variables for a piezo-effect are, respectively, stress and charge. Most variables are associated with energy, e.g., pressure, displacement, velocity, temperature, charge, current, etc., while some are properties of the system components that may undergo change due to a physical effect. For instance, resistance of a resistor is a variable in the context of its susceptibility to change with temperature, and is the output of the temperature-resistance effect. *Properties* are the characteristics that together specify the generic concept of a component. A *component* is an object that has a set of properties that can help acti-

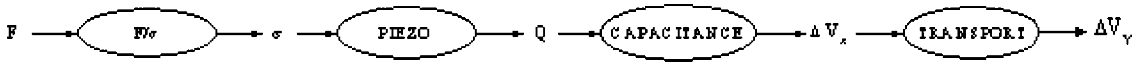

Fig. 2 Representation of a solution principle

vate effects, e.g., a ferromagnetic bar is a component with the following properties: bar (prevalent dimension and cross-sectional area), ferromagnetic, solid, surface, electrical conductor and heat conductor. *Constraints* are relationships that are assigned between a component and other components, or between a component and the reference frame. These relationships can be geometrical, spatial or among component properties. While properties are typical of a component, and owned independently of the particular structure of which it is a part, constraints are attributes that a structure expresses because the components it is made of are organised in a particular arrangement or because of the relation between the structure and the larger framework within which it resides. Each *effect* is capable of transforming some inputs to some outputs. In order to perform this transformation, an effect needs some attributes (properties and constraints) to be present in the context in which it operates. A product *function* is represented as a transformation between input and output variables (e.g., a force input to be measured by an output voltage for the above sensor).

A sensor at the *solution principle* level is represented as a combination of effects that are strung together using their input and output variables. For instance, the piezo-electric sensor is represented as a chain of four effects as shown in Fig. 2. A sensor at the *conceptual structure* level is represented as a combination of components that are interfaced with adjacent components; some of the components may be constrained using appropriate constraints. For instance, the above sensor has a conceptual structure with three components and a constraint: a piezo-crystal that is constrained against movement, conductor plates that are interfaced with the piezo-crystal, and cables that are interfaced with the conductor plates (Fig. 3).

6 Reasoning

In order to enable automated synthesis of sensor solution principles for a given intended function, and of alternative conceptual structures for a given solution principle, two databases are devised. One is a database of effects that links each effect with its input and output variables and the properties and constraints needed for the effect to be activated. For example, the force-stress effect has stress as output for force as input (variables); this requires a solid surface (property) constrained against movement (constraint), see Fig. 4.

The second database links components with their properties and allowable variables. For instance, a piezo-crystal component has the properties of a solid with surfaces having dielectric, piezo-crystal and low heat-conduction properties, and has the ability to conduct force, stress, strain, electrical charge and voltage, in addition to being susceptible to stray magnetic and electrical fields. The existence of variables in the representation of intended functions and effects allows solution principles, which are combinations of effects, to be generated to fulfil a given intended functionality. The existence of properties in both the databases allows the effects in a given principle to be replaced by components, enabling automated synthesis of conceptual structures for the solution principle. The structure sharing algorithm has three steps: synthesis of solution principles for a given intended function, synthesis of initial conceptual structures for a given solution principle, and integration of a given initial conceptual structure into a structure shared one if possible.

Fig. 3 Representation of a conceptual structure

Synthesis of solution principles (see [6] for detail) starts by identifying a list of effects, from the effects-database, which have the same input variable as that of the intended function. For each such effect, its output variable is identified, and checked against the output variable of the intended function. If the two match, that effect can act as a solution principle for the problem. Otherwise, the output of the effect identified is set as the input variable for the next iteration, and the above procedure repeated, which lead to stringing together of two effects. This is done until the number of effects strung together exceeds a number pre-specified by the designer. The outcome is an exhaustive list of solution principles, each having the overall input/output as specified in the intended function.

Synthesis of conceptual structures of a given solution principle is done by first identifying the list of properties and constraints required for each effect in the solution principle, see Fig. 5. The components database is then searched to find all possible component alternatives that can satisfy each of these properties. Each combination of components, one for each property necessary, forms an alternative initial conceptual structure (Fig. 6).

An initial structure consists of a list of components, each of which satisfies only one of the properties required by the solution principle. *Integration of the an initial structure* is done now by identifying each component in the initial structure that is chosen more than once, consolidating the copies into a single component, and propagating the constraints and interfaces accordingly. For instance, the initial conceptual structure in Fig. 6 shows that a piezo-crystal is used to activate the force-stress effect by constraining it against movement, another to activate piezo effect, and yet another, together with conductor plates, to activate capacitance effect. Also, the first crystal has to be adjacent to the second, while the second is to be adjacent to the third while also being adjacent to the conductor plates. Therefore, consolidation of the crystals is done by replacing these three crystals by one, and propagating constraints and interfaces to ensure that it is constrained against movement and remains adjacent to the conductor plates (which are adjacent to the cables). The structure shared final structure is shown in Fig. 3.

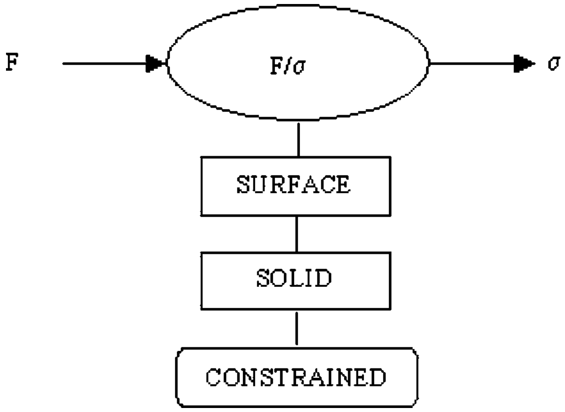

Fig. 4 An effect linked to its I/O variables, and properties and constraints necessary

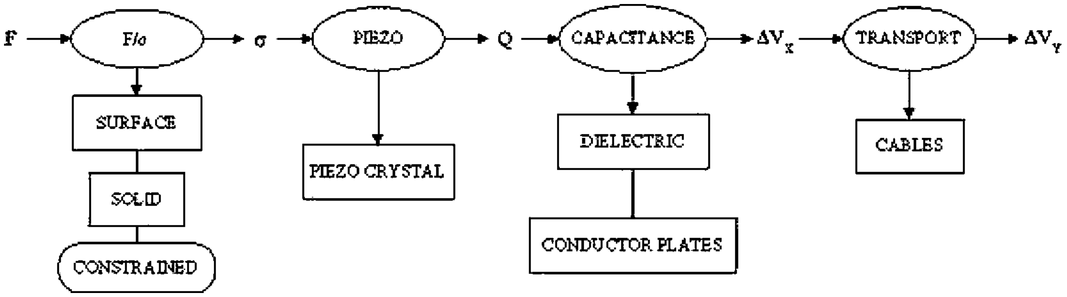

Fig. 5 A solution principle (chain of ovals) linked to the properties and constraints necessary (boxes)

7 Implementation

The approach is implemented into a computer program with CommonLISP as the implementation language. The code is currently suitable only for SISO (Single Input Single Output) sensor chains of effects. However, a wide variety of sensors fall into this category, which indicates the importance of supporting synthesis of this class of sensors: being simple they are commonplace and constitute a majority of the sensors developed.

In implementation, an effect is represented as a list with the following attributes:

Effect Name:	Name of the effect (e.g., Peltier effect)
Input:	Input for activating the effect (e.g., voltage)
Output:	Output from the effect (e.g., heat)
Properties:	Required properties (e.g., two metals in contact at two points)
Constraints:	Constraints on properties (e.g., the metals must be dissimilar)

A component is represented as a list with the following attributes:

Component Name:	Name of the effect (e.g., Ferromagnetic bar)
Properties:	Properties of the component (e.g., bar, solid, surface, ferromagnetic . . .)

A solution principle is represented as a list with the following attributes:

Input:	Input to the principle (e.g., force)
Output:	Output from the principle (e.g., voltage)
Effects:	Ordered list of all effects with I/O, properties and constraints. For instance, the solution principle in Fig. 2 has the following list of effects, I, O, properties and constraints: (F/sigma F sigma (solid surface) constrained) (Piezo sigma charge (piezo-crystal) nil) (Capacitance charge voltage (dielectric conductor-plates) nil) (Transport voltage voltage (cables) nil).

A conceptual structure is represented as a list with the following attributes:

Components:	Ordered list of components and constraints. For instance, for the conceptual structure in Fig. 3, the list contains these components and constraints: (piezo-crystal constrained) (conductor-plates nil) (cables nil).
Effects:	Effects with components replacing properties they satisfy. For instance, for the above conceptual structure, this list contains the following: (F/sigma F sigma (piezo-crystal piezo-crystal) constrained) (Piezo sigma charge (piezo-crystal) nil) (Capacitance charge voltage (piezo-crystal conductor-plates) nil) (Transport voltage voltage (cables) nil).

In order to generate solution principles for a given product function, and conceptual structures to a given solution principle, the software takes the following user inputs: the expected sensor input and output, the maximum allowable number of effects in a solution principle, and the maximum allowable number of components in a conceptual structure.

The program first generates a list of all the possible solution principles that can be constructed by composing effects available in the effects database. For any principle chosen among these, it then generates all its possible conceptual structures that can be constructed by composing components available in the components database. For a given solution principle, the proposed structure sharing approach creates possible conceptual structures, structure shared where possible, using the following steps:

1. *Identify all properties required for activation of each effect in the given principle*. For instance, in the case of the solution principle in Fig. 2, the effects and corresponding properties necessary are: (F/sigma (solid surface)) (Piezo (piezo-crystal)) (Capacitance (dielectric conductor-plates)) (Transport (cables)). If

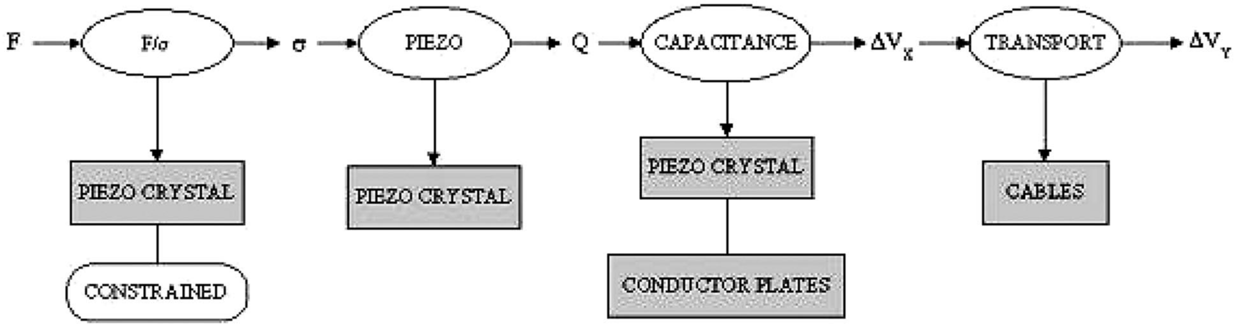

Fig. 6 A solution principle linked to an initial conceptual structure (boxes in gray) and a constraint

Table 1 Comparison of existing solution principles with those generated

case type	I	O	no. of effects	no. of SP	no. of SP synthesised	no.(new) /no.(exi)
Piezo-elec.	force	volt.	4	1	1	0/1
Seebeck	tem-d.	volt	2	1	1	0/1
Resistance	temp.	volt	3	1	2	1/1
Resistance	pres.	volt	3	1	1	0/1
Capacitance	pres.	volt	3	2	3	1/2
Strain Gauge	strain	volt	4	2	2	0/2
Thermistor	strain	volt	4	2	2	0/2
Potentiometer	disp.	volt	3	2	2	0/2
Magnetostrict.	force	volt	5	1	1	0/1
Capacitance	disp.	volt	2	1	1	0/1
Self-Inductance	speed	volt	3	1	2	1/1
Nozzle-Flapper	disp.	pres.	3	1	1	0/1
Pressure	temp.	disp.	3	1	1	0/1
Ammeter	curr.	volt	4	1	1	0/1
Pressure	weight	pres.	2	1	1	0/1

there are e effects in a principle, and if each requires an average of p properties to be activated, a total of $p.e$ properties must be present for the principle to be activated.

2. *Identify all the components, from the components database, that can provide each of the properties necessary*. For instance, for the capacitance effect in the above solution principle, the database provides three component alternatives that have dielectric property: dielectric-solid, dielectric-fluid and piezo-crystal. For an average of c alternative components having each property, there are $p.e$ lists of c components each.

3. *Concatenate a component each from the lists to generate alternative conceptual structures*. The number of possible alternative conceptual structures is $c^{p.e}$.

4. *Reduce the number of components to minimum in each structure alternative by deleting multiple occurrences of the same component to one in the structure*. For instance, the components, in one possible conceptual structure, providing appropriate properties to embody the effects in the solution principle in Fig. 2, shown in bold in the list ((F/sigma F sigma (**piezo-crystal piezo-crystal**) constrained) (Piezo sigma charge (**piezo-crystal**) nil) (Capacitance charge voltage (**piezo-crystal conductor-plates**) nil) (Transport voltage voltage (**cables**) nil)), get reduced to only three components: piezo-crystal, conductor-plates and cables. The number of operations is of the order of $c^{p.e}$.

The estimated number of operations necessary for this approach is given by: $\mathrm{N(proposed)} = p.e + p.e.c + 2.c^{p.e}$ (See Appendix 1 for more detail).

This should create all possible alternative conceptual structures with various degrees of structure sharing, including the optimally structure shared ones, and is more efficient than the existing approach [1], for which the estimated number of operations is given by: $\mathrm{N(existing)} > p.e + p.e.c + [c^{pe}\{(2pe)^{0.5}\}^{0.5pe}]$, see Appendix 1 for more detail for derivation of these equations, and comparison of efficiency of the two approaches.

8 Testing and Evaluation

Data collected on 15 families of SISO sensors are used to evaluate the approach. These include the 11 cases of data obtained at the beginning of the project as well as 4 more cases collected for the express purpose of evaluation. The effectiveness of the approach is evaluated by using the program to generate a list of solution principles for each of these cases followed by generation of all the conceptual structures generated for each existing solution principle. These are then compared with the data collected. The objectives have been to see whether the set of solution principles and conceptual structures generated by the program (1) includes the solution principles and conceptual structures existing in the data, and (2) contains other, novel, realisable principles and structures.

Table 1 shows the results of comparison between the existing principles and those generated by the synthesis software for which at least one conceptual structure exists. The column headings are as follows. The 1st column indicates the type of principles involved in the existing cases considered, where each case considers a family of sensors. The 2nd and 3rd columns respectively specify the input and output required by the intended function of the sensors, where *tem-d.*, *pres.*, *volt.*, *curr.*, *temp.* and *disp.* mean temperature difference, pressure, voltage, electric current, temperature and displacement respectively. The 4th column specifies the number of effects that the program was allowed to use to generate the solution principles in each case (abbreviated as SP). Column five gives the number of alternative solution principles that were identified from the existing data. Column six gives the number of solution principles synthesised by the program in each case. Column seven (last column) gives the ratio of the number of principles proposed by the program that are novel to those proposed that already existed in the data analyzed.

Column seven shows whether the program can predict the solution principles of the existing sensors, which it does in each case, and whether it creates any new, realisable (i.e., having at atleast one conceptual structure) solution principle for the same function, which it does in 6 out of 15 cases. Note that in some cases the number of solution principles generated is different from each other even though the input, output and the maximum number of principles allowed in each case is exactly the same (e.g., rows 4 and 5 in Table I). This is due to the fact that the maximum number of components allowed in a conceptual structure in each case is different from one another. Since the focus of generating solution principles is primarily to see whether any alternatives to the existing conceptual structure can be created that would use the same or less number of components than the existing structures, the maximum number of components allowed in a case is taken to be the one used in its existing sensor.

The results of the comparison between the existing conceptual structures and those generated by the software are shown in Table 2. The column headings are similar to those in Table 1, except for column eight, which gives the ratio of how many structures synthesised are structure shared to the total number of structures synthesised.

Column 7 shows whether the program can predict the conceptual structures of the existing sensors, which it does in each case, as the number of existing structures (Column 5) is always the same as the denominator in the ratio in Column 7. In 7 of the 15 cases, it generates other, alternative conceptual structures for the sensors (shown by non-zero values of the numerator in the ratio). Except for three cases where even the existing conceptual structures are not structure-shared, the program generates structure-shared structures in each case. This demonstrates the generality and power of the approach in creating structure-shared designs

Table 2 Comparison of existing conceptual structures with those generated

case type	I	O	no. of comp.	no. of CS	no. of CS synthesised	no.(new) /no.(exi)	no(SS) /no(syn)
Piezo-elec.	force	volt.	3	1	1	0/1	1/1
Seebeck	tem-d.	volt	2	1	7	6/1	7/7
Resistance	temp.	volt	2	1	1	0/1	1/1
Resistance	pres.	volt	2	1	1	0/1	1/1
Capacitance	pres.	volt	3	1	1	0/1	0/1
Strain Gauge	strain	volt	3	1	1	0/1	1/1
Thermistor	strain	volt	3	1	1	0/1	1/1
Potentiometer	disp.	volt	3	1	1	0/1	1/1
Magnetostrict.	force	volt	3	1	1	0/1	1/1
Capacitance	disp.	volt	3	1	3	2/1	0/3
Self-Inductance	speed	volt	3	1	2	1/1	0/2
Nozzle-Flapper	disp.	pres.	3	1	4	3/1	4/4
Pressure	temp.	disp.	2	1	2	1/1	2/2
Ammeter	curr.	volt	5	1	16	15/1	4/16
Pressure	weight	pres.	2	1	2	1/1	0/2

whenever possible. The number of conceptual structure alternatives generated, however, are often few. However, as the number of components allowed (Column 4) is taken as the minimum that would generate any conceptual structures at all (except in Case 14), increasing this would generally increase the number of alternative conceptual structures, e.g., in Case 14, using the minimum allowable number of components of 4 would have given 4 alternative conceptual structures.

There are a number of cases where the conceptual structure alternatives generated by the program are quite interesting. For instance, in the Seebeck effect thermometer case, the existing design activates Seebeck effect using two dissimilar metals connected at two junctions, the change in temperature between which sets up a voltage that is transported using cables. The conceptual structure of this existing sensor is given in Fig. 7, where two dissimilar metals are connected together (using DIFFMETAL constraint) and are connected to cables for transport. A novel structure alternative, generated by the program, uses the metal in the existing cable with another different metal to activate Seebeck effect, see Fig. 8. In another alternative, two cables having dissimilar metals are used both to activate Seebeck effect and to transport the voltage across, see Fig. 9. Both these alternatives should be better structure shared that the existing structure. Take the Ammeter case as another example. Here, the existing design (Fig. 10) uses a bar to transmit torque generated in a permanent magnet moving in a static coil due to the input current, and a separate spring to provide the torsional resistance necessary. In contrast, an alternative, novel structure, suggested by the program (see Fig. 11) uses a single bar to do both the transmission and the spring action, thereby reducing the number of components necessary. It is envisaged that if the component database is extended with components other than those used in the existing sensors only, the program should generate many other attractive structure alternatives, with a greater likelihood of some of these being better structure-shared than the existing conceptual structures.

9 Discussion, Conclusions and Future Work

The approach presented is a crucial departure from the only existing approach to computational structure sharing [1]. The existing approach starts with the intended functions of a product, develops a principle as a combination of effects, and replaces each effect by elementary structures. It then deletes some of these structures, and checks to see if the additional properties of the rest of the structures can fulfil the functions of the deleted structures. In other words, the approach progresses through *functions to principles to structures to properties*, in order to achieve structure sharing. In contrast, the proposed approach starts with the intended functions, and synthesises solution principles as combinations of physical effects that satisfy these functions. It then identifies the properties essential for fulfilment of the effects in a principle, and from these creates possible conceptual structures for the principle by combining elementary structures that have the required properties. Finally, it compares the elementary structures in a conceptual structure to identify whether the same elementary structure is used more than once in the conceptual structure. In that case, all these elementary structures are replaced by just one that would now satisfy all these functions. In other words, the approach progresses from *function to principles to properties to structures*, property being the link between principles and structures, as proposed in some earlier studies [2,22].

Besides having the level of representation appropriate for the present task, there are three potential advantages of the approach proposed over the earlier approach. The first is, while the earlier approach encourages structure sharing of a given concept along the lines of adaptive variation, the new approach encourages the more desirable generative generation of alternative structures, allowing choice between resource-effectiveness and changeability. The second advantage is its relative efficiency over the earlier approach. While the earlier approach may have to carry out an

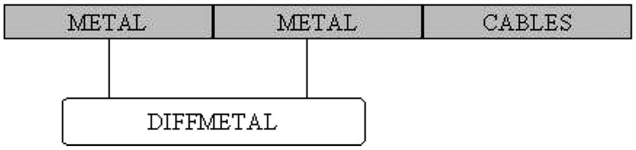

Fig. 7 An existing conceptual structure for the Seebeck effect thermometer

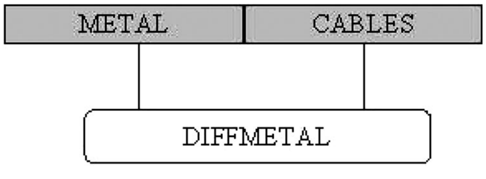

Fig. 8 A new, computer generated alternative conceptual structure for the Seebeck effect thermometer

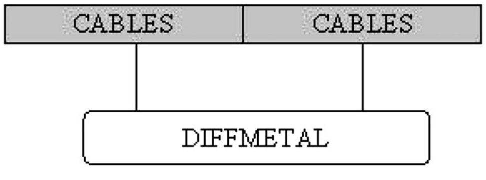

Fig. 9 Another computer generated alternative conceptual structure for the Seebeck effect thermometer

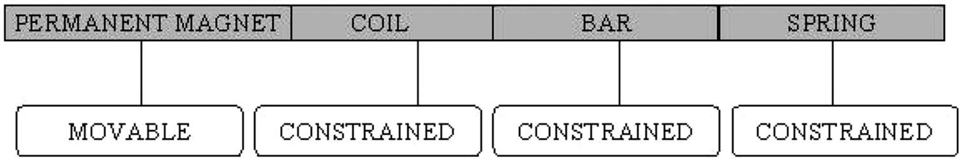

Fig. 10 An existing conceptual structure for the Ammeter case

inefficient iteration between deleting structures, checking ability of other structures to fulfil the functions of the deleted structures and making up for the deleted structures, the new approach allows all these to be done in a direct, non-iterative way. The third advantage is that the proposed approach makes it easier to identify potential side effects in a conceptual structure, [7,8], since it connects physical effects to elementary structures using properties necessary.

The main conclusions are:

- Structure sharing is an important concept for effective use of resources, but little is currently available for supporting its use during design. While structure sharing reduces use of resources in a product, it can decrease its changeability. It is one of four kinds of sharing that provides a variety of trade-offs between these two goals.
- A computational approach has been developed that achieves structure sharing following reasoning through functions, principles, properties and structures. The approach encourages development of alternative concepts, is efficient in its directness, and allows easy detection of side effects. The approach has been tested using a range of sensor designs; existing as well as new conceptual structures for these sensors have been suggested by the approach.

The approach works well for the small database of components and effects and for the kind and level of abstraction of the problems for which it is developed. However, currently each component is described using a set of properties, and development of conceptual structures from solution principles is done by simply replacing properties necessary in principles by components having these properties, on a one to one basis. This may require modification by giving priority to properties that are unusual (e.g., dielectric) over those that are commonplace (e.g., solid or surface), and by giving priority to components that can provide more than one property. Further work involves supporting evaluation of conceptual structures, development of physical embodiments, and extending the approach to other possible areas of application using larger databases.

Acknowledgements

This work has been variously supported by EPSRC, UK, MEI, Japan and DST, India. Riccardo Regno helped develop and implement some of the work presented here.

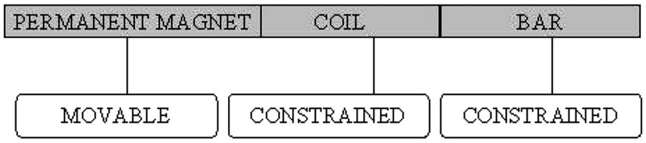

Fig. 11 A new, computer generated alternative conceptual structure for the Ammeter case

Appendix 1: Comparison of Resource-Efficiency of Structure Sharing Approaches

In this appendix, we delineate the steps involved in the two structure sharing approaches, estimate number of computations necessary for developing structure shared conceptual structures in each case, and compare these numbers for comparing the relative resource-efficiency of the two approaches.

For a given solution principle, the structure sharing approach proposed in this paper creates possible conceptual structures, structure shared where possible, using the following steps:

1. Identify all properties required for activation of each effect in the given principle: if there are *e* effects in a principle, and if each requires an average of *p* properties to be activated, a total of *p.e* properties must be present for the principle to be activated.
2. Identify all the components, from the components database, that can provide each of the properties necessary: for an average of *c* alternative components available for providing each property, there will be *p.e* lists of *c* components each.
3. Concatenate a component each from the lists to generate alternative conceptual structures: the number of possible alternative conceptual structures is $c^{p.e}$.
4. Reduce the number of components to minimum in each structure alternative by deleting multiple occurrences of the same component to one in the structure: the number of operations is of the order of $c^{p.e}$.

Hence, the total number of operations necessary in this approach is given by Eq. (1):

$$\mathrm{N(proposed)} = p.e + p.e.c + 2.c^{p.e} \tag{1}$$

This would create all possible alternative conceptual structures with various degrees of structure sharing, including the optimal structure shared ones.

For a given solution principle, the structure sharing approach of Ulrich [1] creates a possible structure-shared structure in the following way:

1. Identify all properties required for activation of each effect in the given principle: if there are *e* effects in a principle, and if each requires an average of *p* properties to be activated, a total of *p.e* properties must be present for the principle to be activated.
2. Identify all the components, from the components database, that can provide each of the properties necessary: for an average of *c* alternative components available for providing each property, there will be *p.e* lists of *c* components each.
3. Create a single conceptual structure by concatenating one component from each list of components: this is taken as a single concatenation operation.
4. Optimize this structure for maximum structure sharing: this is done using the following steps:
   a. Make a list of components that can be deleted: there are *p.e* components.
   b. Delete one of these components and find possible alter-

natives from the remaining components in the structure which could provide the property that was provided by the component deleted: there are between $(c-1)$ and 0 options, since of the c components that could on an average provide a property, one is deleted, and in the worst case none of the remaining components in the structure could provide this property.

c. Choose one of these options as the component now providing the property earlier provided by the deleted component, and list what components can now be deleted: it would be between $(pe-1)$ and $(pe-2)$ since in the scenario where the component deleted cannot be replaced by another component, one has now $(pe-1)$ options, while in the case where the component can be replaced, neither of the component deleted nor its replacement can be deleted again, hence options remaining being $(pe-2)$.

d. Identify what alternative components can be used as replacements: again there are between $(c-1)$ and 0 options.

e. Carry on the process of deletion and replacement until all deletion options are tried out for the list in step 4a.

This approach, however, does not guarantee creation of the optimal solution among all conceptual structures that can be created using a given set of components, but only within the structure chosen in Step 3. In order to optimize this, step 4 has to be continued several times, each time picking a separate conceptual structure. The number of deletion and replacement steps required, in the case where none of the components is replaceable, is *pe*, while that in which all cases require a distinct replacement is $0.5pe$ (since the number of components available for deletion after each deletion and replacement step reduces by twice the number of components available in the first case). The number of alternative structures produced in the first case is 1 (since no reduction of components could be effected). If it is possible to provide $(c-1)$ options in each step of replacement in the second case, the number of alternative structures possible is given by $(c-1)^{0.5pe}$. As the number of steps increases, the number of options available on an average decreases.

The number of operations in the first case, for trying to (internally) optimize a single conceptual structure is given by Eq. (2):

$$\mathrm{N1}=pe \tag{2}$$

Computationally, this is the best case since the number of options to arrive at the conclusions is minimal, as at each deletion step no replacement options are available.

In the second case, the number of operations for trying to internally optimize a single conceptual structure is given by Eq. (3)–(5):

$$\mathrm{N2}=pe.(c-1).(pe-2).(c-1)(pe-4)(c-1)\ldots(\text{having }\ 0.5pe\ \text{ terms}) \tag{3}$$

$$=\{pe(pe-2)(pe-4)\ldots 2\}.(c-1)^{0.5pe} \tag{4}$$

$$>[\{(2pe)^{0.5}\}.(c-1)]^{0.5pe} \tag{5}$$

This is since each multiplication of corresponding terms from either side of the series $\{pe(pe-2)(pe-4)\ldots 2\}$, i.e., *pe.2*, $(pe-2).4$, etc} is not less than $2pe$ (see proof in the paragraph below), thereby average value of the a term in the series being no less than $(2pe)^{0.5}$, and there being $0.5pe$ terms gives the value of the first part of Eq. (4) as greater than $\{(2pe)^{0.5}\}^{0.5pe}$.

That $\{pe(pe-2)(pe-4)\ldots 2\}$ is greater than $(2pe)^{0.5}$ can be proved as follows:

As one goes from left to right in the series, the terms decrease by 2 from the previous term. In other words, the highest term in the series is *pe* while the lowest term is 2. If corresponding terms from either side is multiplied to each other (e.g., *pe.2*, $(pe-2).4$, etc) the value of the products is such that their sum is constant, and in this case equal to $(pe+2)$. We are interested in finding the least value of the product of two corresponding terms from either side in the series, because the value of any term in the series will be greater than the square root of that least value of the product.

As one progresses from outside to inside of the series, taking products of terms from either end, the difference between the products get increasingly less. In other words the largest difference in value of the corresponding terms is for the first and the last term in the series. Let any two corresponding terms in the series be described by variables *x* and *y*, and their difference be denoted by another variable d. Now, we can write down the following relationships between *x* and *y*:

$$x+y=(pe+2) \tag{6}$$

$$x-y=d \tag{7}$$

The values of *x*, *y* and *xy* can be derived as:

$$x=0.5(pe+2+d) \tag{8}$$

$$y=0.5(pe+2-d) \tag{9}$$

$$xy=0.25\{(pe+2)^2-d^2\} \tag{10}$$

Since the value of *xy* is minimum when the value of $|d|$ is maximum, minimum value of xy is when the difference in value between the two corresponding terms is maximum, and is given by $2pe$.

Now, assuming that on an average the number of operations will be between the above two extreme cases (i.e., N1 and N2), the average number of operations required for trying to optimize each conceptual structure is given by the following:

$$\mathrm{N}(av)=0.5[pe+\{(2pe)^{0.5}.(c-1)\}^{0.5pe}] \tag{11}$$

$$>0.5\{(2pe)^{0.5}.(c-1)\}^{0.5pe} \tag{12}$$

In one such optimization attempt, an average of $0.5\{1+(c-1)^{0.5pe}\}$ conceptual structures will be already considered. Therefore, the number of attempts necessary to guarantee finding the overall optimal structure-shared conceptual structure is given by Eq. (13):

$$n(\max)>c^{pe}/0.5(c-1)^{0.5pe} \tag{13}$$

If in each case, the number of operations necessary is the average found in Eq. (12), the total average number of operations necessary before the overall optimum can be found is given by Eq. (13)–(14):

$$\mathrm{N(existing)}>p.e+p.e.c+[c^{pe}/0.5(c-1)^{0.5pe}].[0.5\{(2pe)^{0.5}(c-1)\}^{0.5pe}] \tag{14}$$

$$>p.e+p.e.c+[c^{pe}\{(2pe)^{0.5}\}^{0.5pe}] \tag{15}$$

Therefore, the ratio of number of operations in existing and proposed approaches is given by Eq. (16):

$$\mathrm{N(existing)/N(proposed)} >[p.e+p.e.c+c^{pe}\{(2pe)^{0.5}\}^{0.5pe}]/[p.e+p.e.c+2.c^{p.e}] \tag{16}$$

The ratio of values of the last terms in the numerator and denominator will govern the value of the ratio, which is given by Eq. (17)–(18):

$$\mathrm{N(existing)/N(proposed)}>[c^{pe}\{(2pe)^{0.5}\}^{0.5pe}]/2.c^{p.e} \tag{17}$$

$$>0.5\{(2pe)^{0.5}\}^{0.5pe} \tag{18}$$

As long as $0.5\{(2pe)^{0.5}\}^{0.5pe}$ is not less than 1, existing approach on an average would require more operations than the proposed approach, and therefore less efficient computationally.

This is true for any value of pe greater 2, which is the least number of elements necessary for any structure sharing to be possible.

References

[1] Ulrich, K., 1988, ''Computational and Pre-Parametric Design,'' PhD Thesis, Artificial Intelligence Laboratory, MIT Cambridge.

[2] French, M., 1992, *Form, Function And Mechanism*, McMillan, London, pp. 11–13.

[3] Whitney, D., 1996, ''Why Mechanical Designs Cannot be Like VLSI Design,'' Res. Eng. Des., **8**, pp. 125–138.

[4] Chakrabarti, A., 2001, ''Sharing in Design: Categories, Importance and Issues,'' *Proc. Intl. Conf. on Eng. Design (ICED01)*, pp. 563–570, Glasgow.

[5] Sessler, G. M., 1994, ''Silicon Microphones,'' *Proc. 27th Meeting Acoustical Society of America*, MIT, Cambridge, MA.

[6] Chakrabarti, A., Johnson, A., and Kiriyama, T., 1997, ''An Approach to Automated Synthesis of Solution Principles for Micro-Sensor Designs,'' *Proc. Intl. Conf. on Eng. Design*, Tampere, **2**, pp. 125–128.

[7] Chakrabarti, A., and Johnson, A. L., 1999, ''Detecting Side Effects in Solution Principles,'' *Proc. Intl. Conf. on Eng. Design*, Munich, **2**, pp. 661–666.

[8] Chakrabarti, A., and Regno, R., 2001, ''A New Approach to Structure Sharing,'' *Proc. Intl. Conf. on Eng. Design*, pp. 155–162, Glasgow.

[9] Chakrabarti, A., 1991, ''Designing by Functions,'' Univ. of Cambridge, Cambridge.

[10] Chaplin, R. V., Li, M., Oh, V. K., Sharpe, J. E. E., and Yan, X. T. 1994, ''Integrated Computer Support for Inter-Disciplinary System Design,'' *Proc. AI in Design Conf.*, J. Gero, and F. Sudweeks, Eds., Kluwer Academic, Doredrecht.

[11] Sushkov, V., Alberts, L., and Mars, N. J. I., 1996, ''Innovative Design Based on Sharable Physical Knowledge,'' J. S. Gero and F. Sudweeks (Eds), *Proc. AI in Design Conf.*, Kluwer Academic, Dordrecht, pp. 723–742.

[12] Zavbi, R., and Duhovnik, J., 1997, ''Prescriptive Model With Explicit Use of Physical Laws,'' *Proc. Intl. Conf. on Eng. Design (ICED97)*, Tampere, pp. 37–44.

[13] Zavbi, R., and Duhovnik, J., 2000, ''The Problems of Transition from Basic Schematics to a Schematic of a Technical System,'' *Proc. Intl. Design Conf.-Design 2000*, Dubrovnik, pp. 67–72.

[14] Welch, R. V., and Dixon, J. R., 1994, ''Guiding Conceptual Design Through Behavioral Reasoning,'' Res. Eng. Des., **6**(3), pp. 169–188.

[15] Williams, B. C., 1989, ''Invention from First Principles Via Topologies of Interaction,'' PhD Thesis, MIT, Cambridge.

[16] Roth, K., 1970, ''Systematik Der Machinen Und Ihrer Mechanischen Elementaren, Funktionen,'' Feinwerktechnik, **74**, pp. 453–460.

[17] Agarwal, M., and Cagan, J., 1999, ''Systematic Form and Function Design of MEMS Resonators Using Shape Grammars,'' *Proc. Int. Conf. on Eng. Design*, Munich, **2**, pp. 823–828.

[18] Hammond, K. J., 1986, ''CHEF: A Model of Case Based Planning,'' Proc. AAAI-86, Philadelphia, PA, pp. 267–271.

[19] Pu, P., and Purvis, L., 1994, ''Formalizing Case Adaptation in a Case Based Design System,'' *Proc. AI in Design Conf.*, J. S. Gero & F. Sudweeks, Eds., Kluwer Academic, Doredrecht, pp. 77–91.

[20] Andreasen, M. M., Hansen, C. T., and Mortenssen, N. H., 1996, ''The Structuring of Products and Product Programmes,'' *Proc. 2nd WDK Workshop on Product Structuring*, M. Tichem, T. Storm, M. M. Andreasen and K. J. MacCallum eds., TU Delft, Netherlands.

[21] Hansen, C. T., 1997, ''Towards a Tool for Computer Supported Structuring of Products,'' *Proc. Intl. Conf. on Eng. Design*, Tampere, **2**, pp. 71–76.

[22] Ringstad, P., 1999, ''The Transition Between Concept and Layout—Early Component Design,'' *Proc. Intl. Conf. on Eng. Design*, Munich, **2**, pp. 1171–1174.

[23] Fricke, G., 1992, ''Experimental Investigation of Individual Processes in Engineering Design,'' *Proc. Research in Design Thinking*, N. Cross, K. Dorst and N. F. M. Roozenburg, eds., Delft University Press, Delft, pp. 105–109.

[24] Bentley, J. P., 1983, *Principles of Measurement Systems*, Longman, NY.

[25] Cooper, W. D., and Helfrick, A. D., 1985, *Electronic Instrumentation and Measurement Techniques*, Prentice-Hall international, New Jersey.

[26] Sydenham, P., 1985, *Transducer In Measurement and Control*, Adam Hilger, Bristol.