A Formal Functional Representation Methodology for Conceptual Design of Material Flows-Processing Devices

Yong Chen[a*], Meng Zhao[a], Ying Liu[b], Youbai Xie[a]

[a] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China

[b] Institute of Mechanical and Manufacturing Engineering, School of Engineering, Cardiff University, UK

* Corresponding Author: Yong Chen

Add.: Room 801, School of Mechanical Engineering, Shanghai Jiao Tong University

No. 800, Dongchuan Road, Minhang District, Shanghai, 200240, China

Tel: +86-153-0160-6712;

Email: mechenyong@sjtu.edu.cn;

Page Number	40
Figure Number	6
Table Number	0

A Formal Functional Representation Methodology for Conceptual Design of Material Flows-Processing Devices

Abstract

Although there has been considerable computer-aided conceptual design research, most of the proposed approaches are domain-specific and can merely achieve conceptual design of energy flows-processing systems. Therefore, this research is devoted to the development of a general (i.e., domain-independent) and knowledge-based methodology that can search in a wide multi-disciplinary solution space for suitable solution principles for desired material flows-processing functions without designers' biases towards familiar solution principles. It first proposes an ontology-based approach for representing desired material flows-processing functions in a formal and unambiguous manner. Then a rule-based approach is proposed to represent the functional knowledge of a known solution principle in a general and flexible manner. Thereafter, a simulation-based retrieval approach is developed, which can search for suitable solution principles for desired material flows-processing functions. The proposed approaches have been implemented as a computer-aided conceptual design system for test. The conceptual design of a coins-sorting device demonstrates that our functional representation methodology can make the proposed computer-aided conceptual design system to effectively and precisely retrieve suitable solution principles for a desired material flows-processing function.

Keywords: Conceptual Design; Function; Solution Principle; Material Flows-Processing Devices

1 Introduction

According to Pahl & Beitz (1997), a critical task of conceptual design is to search for suitable solution principles for desired functions. Hereby, a solution principle (also called principle solution) of a desired function can be regarded as a basic physical mechanism for achieving the desired function. During conceptual design, it is encouraged that designers should represent a desired function in a solution-neutral way, and search in a wide multi-disciplinary (i.e., cross-domain) solution space for novel and promising solution principles. However, this can often pose a big challenge for engineering designers, since they are often taught with limited multi-disciplinary solution knowledge in a specific major.

A possible approach to addressing the above challenge is to develop a Computer-Aided Conceptual Design (abbreviated as "CACD" later) system, which can search in a multi-disciplinary solution space for generating suitable solution principles. Note that the CACD approach to be developed here should be different from most of the traditional CACD approaches, since it should be independent of any specific domains and be able to search in a wide cross-domain solution space. In contrast, the traditional CACD approaches, which are often based on the expert system methodology, have employed domain-specific representations and reasoning rules to generate solution concepts for energy flow-processing systems (e.g., Welch & Dixon, 1994; Chakrabarti & Bligh, 1996; Campbell et al., 2000; Chen et al., 2006; Kurtoglu, 2009; Helms et al., 2013). For example, based on the bond graph methodology (Rosenberg, 1983), Welch & Dixon (1994) employ a 7-ary tuple for representing the electromechanical flows and rely on a set of transformation rules to generate electromechanical solution concepts. It can also be found that little research has been carried out to achieve conceptual design of Material Flows-Processing (abbreviated as "MFP") devices, which often deals with various domains and thus requires a domain-independent CACD approach. An exception is the feature-based design catalogue research by Feng et al. (1996), where classifications of function verbs and material flows have been developed to index solution principles for retrieval. Similar classifications of MFP functions also appear in the reconciled functional basis (Hirtz et al., 2002). A major drawback of such functional classifications is that they are relatively rough, and cannot explicitly represent what a MFP function is, which can make a CACD system unable to precisely search for feasible solution principles for a desired MFP function. For example, based on the reconciled functional basis, the function, to refine salt (i.e., to separate sand from salt), should be roughly represented as to refine solid; since there can be many solution principles in the knowledge base that can achieve the function of refining solid, it is then difficult for a CACD system to rule out those infeasible solution principles that can also refine solid. Another drawback of the classification-based representation approach is that it employs an informal verb (e.g., refine), rather than a formal flow, state or relation change, to represent a MFP function. As a result, it would be difficult for a CACD system to employ such verbs to effectively achieve functional reasoning. Therefore, there is still no general (i.e., domain-independent) CACD approach that can effectively search in a wide cross-domain solution space for fulfilling conceptual design of MFP devices, which, though, are very common in technical products.

Therefore, this paper will develop a general and knowledge-based approach for the conceptual design of material flows-processing devices. Since such devices are often complex, it would be very difficult to take a first principle-based approach to address the above issue. Therefore, this research will employ the case-based reasoning methodology to achieve conceptual design of MFP devices. Hereby, case-based reasoning can be broadly construed as a methodology of reasoning from past experiences to generate solutions or explanations to the current problems (Kolodner, 1993). This paper will be focused on how to represent a desired MFP function, how to represent the MFP function of a known solution principle in a material flow-free manner, and how to retrieve suitable solution principles for a desired MFP function. Note that the research introduced in this paper is not a replacement for existing CACD research, but a complement for the current solution principle-generating research.

This paper is organized as follows. Section 2 reviews the related work. Section 3 proposes a general approach for representing desired MFP functions. Section 4 then develops a formal functional representation approach for indexing existing solution principles. Section 5 illustrates a simulation-based approach for retrieving suitable solution principles for a desired MFP function. With the conceptual design of a coins-sorting device as an example, Section 6 demonstrates how the proposed approaches can work. Finally, Section 7 concludes this paper.

2 Related Work

As a critical product development stage, conceptual design is a complex process that is often composed of the following tasks, e.g., clarifying functional requirements, developing functional structure (also called functional modeling), searching for solution principles, generating combinatorial solutions, validating combinatorial solutions, and selecting the most promising combinatorial solutions. To support these tasks, various CACD approaches or tools have been developed in recent decades, e.g., the function-behavior-structure model for representing and reusing solution concepts (Gero, 1990), the graph grammar-based machine design algorithm (Schmidt & Cagan, 1997), the agent-based approach for generating design configurations (e.g., Campbell et al., 2000), the morphological matrix-based approach for design concept synthesis (Chen et al., 2006; Arnold et al., 2008), the SysML-based approach for functional modeling (Wölkl & Shea, 2009), the physics-based validation of functional structures (Sen et al., 2013), the function-based design verification (Deng et al., 2000). Due to limited space, it is impossible to review all kinds of CACD research here. Interested readers can find more CACD research in some recent review papers (e.g., Cagan et al., 2005; Eisenbart et al., 2013). According to our research aim, the following review will be focused on the functional representation approaches in the existing CACD research. In addition, we also briefly analyze how the existing functional representation approaches can be used in the CACD research.

As mentioned before, there are primarily three kinds of functional representation approaches. The first kind is the verb-noun-phrase-based approaches, which come from the value engineering research and employ the natural language-based verb-noun pairs to describe functions (Miles, 1972). This kind of functional representation approaches has been widely used in earlier CACD research (e.g., Qian & Gero, 1996; Umeda et al., 1996; Sturges et al., 1996). In such CACD research, functional representations are more suitable for storing the design intents of existing designs for reuse. Since computers are not good at understanding natural language, such functional representations can merely support the keywords-based solution retrieval. When the verb-noun-phrase-based approaches were used to represent MFP functions, the CACD systems would be very poor at retrieving suitable solution principles.

The second kind is the input-output-flow-based approaches. Much CACD research has employed some tuples to represent the input and output energy flows in mechanical or electromechanical domains, e.g., Ulrich & Seering (1989), Welch & Dixon (1994), Chakrabarti & Bligh (1996), Campbell et al. (2000), Chen et al. (2006). An advantage of the input-output-flow-based representation consists in that it allows a CACD system to employ some domain-specific transformation rules to achieve automated functional reasoning. However, since the tuple-based flow representations are finely tuned to represent mechanical or electromechanical energy flows, they are not general and flexible to represent material flows. Note that different kinds of material flows often have different sets of attributes, which thus cannot be represented with a unified tuple. For example, a water flow may have the attributes evaporability and fluidity, while a salt flow may have the attribute water-solubility; it is impossible to represent such attributes in the tuples used to represent those energy flows. In addition, it is also impossible to employ input-output-flow-based approaches to formally represent a function that does not deal with explicit flow or state changes from input to output. For example, for the relation transformation-focused function, *to separate sand from salt (i.e., to refine salt)*, a designer is concerned with that the mixed relation between salt and sand should be changed as separated (Chen et al., 2013; Chen et al., 2015), rather than any specific change(s) of sand and salt. It is then self-evident that the existing input-output-flow-based approaches cannot represent such a function in a formal and explicit manner.

The third kind is the classification-based approaches, which employ standard classifications of verbs and flow classes to represent functions (e.g., Stone & Wood, 2000; Hirtz et al., 2002). This kind of representation approaches has been widely used in the functional modeling, functional analysis and design synthesis research (e.g., Szykman et al., 2000; Stone et al., 2005; Sridharan & Campbell, 2005; Li et al., 2010; Kurtoglu et al., 2010; Gu et al., 2012; Sen et al., 2013). As mentioned before, a major drawback of the classification-based approach consists in that the standard flow classes are rough and thus cannot explicitly represent what a material flow is, which can make a CACD system unable to precisely retrieve suitable solution principles for a desired MFP function. In addition, the classification-based representation approaches also cannot formally represent the detailed state changes of the flows and their relation changes from input to output. For example, the function representation, *to separate solid from solid*, cannot indicate what state changes will happen to the related solid flows, and how the relation between the two solid flows will change. Without the knowledge about a desired state or relation change, it is impossible for a CACD system to understand what a desired function is, and to retrieve the right solution principles for the function.

It can be found that the functional representation approaches in the existing CACD research are not suitable for formally representing MFP functions in an explicit manner. As a result, it would be impossible for the existing CACD systems to correctly process the functional information and to precisely retrieve suitable solution principles from a large multi-disciplinary solution space for a desired MFP function. Therefore, it is still necessary to propose a formal methodology for representing MFP functions.

3 An ontology-based approach for representing desired MFP functions

To develop a CACD system, it is desirable to have a formal approach for representing a desired function in an unambiguous manner. It is increasingly acknowledged that ontology can assist engineers in representing knowledge in an explicit and sharable manner (Lenat, 1989; Gruber, 1995; Chandrasekaran et al., 1999). Therefore, this research takes an ontology-based approach for representing a desired MFP function. Since an ontology-based representation deals with not only the object representation, but also the representation of the relations between objects (Gruber, 1995), the approach for representing MFP functions here will involve not only how to represent a material flow (i.e., object), but also how to represent the concerned relation between two material flows, as well as how to formally represent a state or relation change of interest.

3.1 An ontology-based object representation model

An object here refers to a material flow that is included in a functional description and needs to be processed by a solution principle. Unlike the energy flows in the previous CACD research (e.g., Welch & Dixon, 1994), an object (i.e., material flow) has various attributes (e.g., state-of-matter, temperature, water-dissolubility, location, magnetizability, size, density, etc.), and different kinds of material flows may have different kinds of attributes. Therefore, it is impossible to employ a rigid tuple-based approach for representing material flows. This research thus employs a flexible attribute-value approach to represent material flows, where a domain expert is allowed to flexibly assign a set of attributes and values to a specific material flow.

This research classifies the attributes of an object as state attributes and disposition attributes. A state attribute is usually employed to describe the state of an object, such as state-of-matter, temperature and location, which are common attributes for material flows. In contrast, a disposition attribute describes a physical disposition of an object, which represents its potential capability to change from one state into another in a specific environment (Bunge, 1977). For example, salt has the water-solubility attribute, which is a disposition attribute that allows salt to dissolve into water. Therefore, a disposition attribute can indicate what state change can occur to an object, which means it can provide designers with significant clues about what physical principles can be employed to process the object, and thus is very important for the conceptual design of a MFP device. It is the fact that the existing functional representation approaches have largely neglected the disposition attributes of a material flow, resulting in that the existing CACD systems (e.g., Prabhakar & Goel, 1998; Goel, 2013) are unable to precisely judge what physical effects can be employed to achieve desired MFP functions. Therefore, the disposition attributes are also included in the object representation model proposed here.

To represent an object with the flexible attribute-value approach, a designer also needs to assign values to the attributes. According to the value types, the attributes of an object can be classified as enumerable attributes and continuous attributes. An enumerable attribute has enumerable values. For example, the attribute *state-of-matter* may have such enumerable values as *solid*, *liquid*, *gas, etc*. In contrast, a continuous attribute (e.g. the temperature attribute) merely has a numerical value. Since numerical values are not suitable for qualitative reasoning in conceptual design, they are discretized into qualitative values. Enlightened by the qualitative simulation theory (Kuipers, 1986), this research employs a set of landmark values to divide the value range of a continuous attribute into a set of intervals. When a numerical value of the attribute falls into a value interval, then the qualitative value of the attribute can be set as the value interval. Note that the landmark values here are independent of any specific objects, since they should be suitable for representing all possible material flows. With the temperature attribute as an example, some landmark values can be chosen as -273ºC, -40ºC, 0ºC, 50ºC, 200ºC, 500 ºC, 1000 ºC, 2000ºC, 6000ºC, etc., and the value intervals can be described as (-273ºC, -40ºC), (-40ºC, 0ºC), (0ºC, 50ºC), (50ºC, 200ºC), (200ºC, 500ºC), (500ºC, 1000ºC), (1000ºC, 2000ºC), etc. It should be noted that the selection of such landmark values largely depends on engineering experience, which might be a weak point. However, compared with some ambiguous and subjective descriptions (e.g., high, low, hot, warm, cold), such value interval-based descriptions are more explicit and objective.

To sum up, an object (i.e. material flow) can then be represented as a set of attribute-value pairs, i.e., {(*attribute*, *value*)}. The attributes of an object here involve not only state attributes, but also disposition attributes. For example, the material flow in a specific situation, *salt*, can be represented as {(*state_of_matter*, "Solid"), (*temperature*, "(0ºC, 50ºC)"), (*water_solubility*, "(10g, 100g)"), ...}. To facilitate the representation of an object, many possible attributes and their values have been collected and standardized, which serve as a content ontology for our functional representation approach, and a designer can then choose suitable ones from them for representing an object. Note that an attribute-value pair of an object, (*attribute*, *value*), is equivalent to the representation, *attribute*(*Object*) = *value*, which will also be used later.

3.2 An ontology-based relation representation model

There can be various kinds of relations between the objects of interest, such as the parallel relation, the mixed relation, the assembly relation, etc. Such relation descriptions are informal representations, and, therefore, cannot effectively support functional reasoning. To represent MFP functions in a formal manner, it is also desirable to have a formal model for representing the relations between two objects.

Assuming that there are two objects, *shaft A* and *shaft B*, which are in a parallel relation, it can be found that the parallel relation actually means that these two shafts are with the same axial orientation. Therefore, a relation between two objects can be formally represented as:

relation_attribute(Object1, *Object2)* = *relation_value*.

Hereby, *relation_attribute* indicates what attribute the relation is concerned with, and *relation_value* describes the relation that the objects (i.e., *object1* and *object2*) are in with respect to the attribute, which can be set as "=", "≠", ">", "<", ">>", etc. With the mixed relation between salt and sand as an example, their mixed relation can be formally represented as: *location(Salt*, *Sand)* = "=". It can be found that a relation between two objects can also be formally represented with the above ontology-based relation model.

3.3 An ontology-based functional representation approach

Based on the object representation model and the relation representation model proposed above, a 6-nary tuple, (O_{in1}, O_{in2}, R_{in}, O_{out1}, O_{out2}, R_{out}), can then be used to conceptualize a MFP function. Here, O_{in1} and O_{in2} denote two input objects (i.e., material flows) of interest, R_{in} represents the concerned relation between the input objects, O_{out1} and O_{out2} are used to represent the requirements on the output objects, and R_{out} is the desired relation between the output objects. Note that if a MFP function is a flow or state transformation-focused function, the second input object, O_{in2}, and the corresponding output object, O_{out2}, can be omitted.

Based on the above conceptual model, a functional representation schema can then be proposed, as shown in Fig. 1. Note that the functional description in the schema is just an informal remark about the design purpose, which a CACD system will not process during conceptual design. Together with the schema, how to represent the MFP function, *to refine salt* (i.e., *to separate sand from salt*), is also illustrated in Fig. 1 as an example. In this function, the (rough) salt to be refined is represented as two objects (i.e., salt and sand) and the mixed relation between them, while the desired output relation between them is defined as that they should not be in the same location, i.e. $location(Salt_{out}, Sand_{out})$ = "$\neq$".

It can be found the proposed ontology-based approach can formally represent a desired MFP function with the aid of an object-attribute representation and an attribute-relation representation, which is then possible for a CACD system to correctly process a desired function. Especially, the disposition attributes in an object-attribute representation disclose the implicit knowledge about what changes can happen to an object, so that it is possible for a CACD system to employ some of these changes to achieve a desired function.

[Fig. 1. An ontology-based functional representation schema and an example]

4 A rule-based approach for representing functional knowledge

To allow a CACD system to retrieve suitable solution principles, it is also necessary to have a formal approach to representing the functional knowledge of a known solution principle, i.e., what functions can be achieved by a solution principle. Different from the approach proposed in Section 3, the functional knowledge representation here should be general and independent of any specific objects (i.e., material flows), so that the corresponding solution principle can be retrieved to achieve similar functions but with different objects. Note that the functional knowledge representation approach here is different from that developed for achieving model-based reasoning (e.g., Chandrasekaran, 1994), which is concerned with the representation of the internal causal behavior process of an existing device. In this research, the functional knowledge of a solution principle is formally represented as a rule-based schema. The proposed functional representation approach is composed of the precondition knowledge model and the action knowledge model, which are elaborated in details as below.

4.1 The precondition knowledge model

The precondition knowledge model involves a set of preconditions, which specify under what conditions a known solution principle can be used to achieve a desired function. For the solution principle of a MFP system, such preconditions can involve the requirements on the state attributes of the input objects, the requirements on their disposition attributes, as well as the requirements on the relations between them. For example, in order for a water dissolving-based separating solution to work, it is required that the state of the two related objects should be in the state of solid, that the object to be removed should have a good water solubility (a disposition attribute), and that the two input objects should be at the same location (i.e., in the mixed relation), etc.

Note that an attribute requirement of a precondition representation of a known solution principle may deal with multiple allowable values, which is different from the object representation of a desired MFP function that merely has one value for each attribute, since it is possible that a solution principle can act on the objects in different states. For example, a gas oven cannot only heat solid food, but also liquid food, as well as the mixture food. Therefore, an attribute requirement on the input object should be represented as:

$obj_{in}.attribute \in \{allowable\_value\}$

where, $\{allowable\_value\}$ is a set of allowable values for the solution principle with respect to *attribute*.

Similarly, a relation requirement in a precondition representation may also deal with multiple relation values, and thus should be represented as:

$relation\_attribute(Obj1_{in}, Obj2_{in}) \in \{allowable\_relation\_value\}$,

where, $\{allowable\_relation\_value\}$ is a set of allowable relation values.

Based on the above attribute-value model and the attribute-relation model, the functional precondition of a known solution principle can then be explicitly represented in a formal manner. Note that the precondition knowledge model of a solution principle is merely related to some attributes of a specific object, which often has many more attributes than those in the precondition representation.

4.2 The action knowledge model

The action knowledge model elaborates what changes a known solution principle can make to the input objects and the relations between them, if the known solution principle is supposed to act on them. Note that the action knowledge here involves not only the expected change(s), but also the unexpected change(s) that is necessary for achieving the function. For example, in order for a water-dissolving effect based solution to separate sand from salt, the expected (desirable) change is a relation change that sand and salt get separated. However, to achieve this expected change, the solution principle also needs to dissolve salt into water, which is an unexpected change to salt, since this change is not expected by a designer when the designer was defining the desired function in a solution-neutral manner. Since such unexpected changes are also essential to achieve the desired function, both expected changes and unexpected changes should be represented in the action knowledge model of a solution principle. As mentioned before, the actions of a solution principle may involve both attribute changes and relation changes. Therefore, the action representation here should involve two kinds of change representations, i.e., the attribute change representation and the relation change representation.

An attribute change representation refers to the change that a solution principle can make to an attribute of an object, which can usually be represented as a change of the value of the attribute from input to output. The change of the value of the attribute of the object indicates how the solution principle changes the state of the object. For example, state-of-matter is an attribute of salt, and the value of this attribute can be changed from solid to liquid, if a water-dissolving effect-based solution acts on the salt. Note that an input value can also correspond to multiple output values, since a solution principle can be used in different situations to achieve different results. To represent the corresponding relationship between the input value and the output value(s), an attribute change can be further represented as a rule, i.e.,

IF ($obj_{in}.attribute = value_{in}$) THEN ($obj_{out}.attribute \in \{value_{out}\}$).

The above rule means that if the *attribute* of a desired function's input object obj_{in} has the value $value_{in}$, then the value of the corresponding attribute of the output object obj_{out} will be changed into one of the values in the value set $\{value_{out}\}$. Assuming that a solution principle can heat an object (e.g., increase its temperature), if the solution principle has two allowable values regarding the temperature attribute of the input object, i.e., "(0ºC, 50ºC)" and "(50ºC, 200ºC)", then two possible action representations are as below. Some rules as such can then be represented as below,

Rule 1: IF ($obj_{in}.temp$ = "(0ºC, 50ºC)")

THEN ($obj_{out}.temp \in$ {"(50ºC, 200ºC)", "(200 ºC, 500ºC)", ...});

Rule 2: IF ($obj_{in}.temp$ = "(50ºC, 200ºC)")

THEN ($obj_{out}.temp \in$ {"(200ºC, 500ºC)", "(500ºC, 1000ºC)", ...}).

Note that it is also possible that the value of an attribute of an object does not change too much, which may result in that the output value does not exceed the landmark value. In such a situation, the output value of the attribute still has to be assigned with the same qualitative value as the input. In order to indicate the difference between the input value and the output one, an input-output relation can then be used to represent the input-output change, i.e.,

IF ($obj_{in}.attribute = value_{in}$)

THEN {($obj_{out}.attribute = value_{in}$) & ($attribute(obj_{in}, obj_{out}) = relation\_value$)}.

For example, when a microwave oven is used to warm food, the temperature of the output food may not exceed 50 ºC. The action knowledge in such a situation can then be represented as below:

IF ($obj_{in}.temp$ = "(0ºC, 50ºC)")

THEN {($obj_{out}.temp$ = "(0ºC, 50ºC)") & ($temp(obj_{in}, obj_{out})$ = "<")}.

A relation change refers to the change that a solution principle is expected to make to a relation between two objects, which can also be represented as a relation change from input to output. For example, when salt is mixed with sand (i.e., coarse salt), the mixed relation means an equal location relation between salt and sand, and a dissolving effect based solution can change the relation to unequal. Similar to an attribute change, a relation change can also be represented in a form similar to a rule, i.e.,

IF ($relation\_attribute<obj1_{in}, obj2_{in}> = relation\_value_{in}$)

THEN ($relation\_attribute<obj1_{out}, obj2_{out}> \in \{relation\_value_{out}\}$).

The above relation change rule means that if the attribute relation indicated by *relation_attribute* between input objects $obj1_{in}$ and $obj2_{in}$ is equal to $relation\_value_{in}$, then the relation value between the output objects will be changed into one of the values in the set $\{relation\_value_{out}\}$. For example, the possible relation change that a separating solution principle is supposed to achieve can be represented as below,

IF ($location<obj1_{in}, obj2_{in}> = "="$)

THEN ($location<obj1_{out}, obj2_{out}> \in \{"\neq"\}$).

Based on the above attribute change representation and the relation change representation, the action knowledge of a known solution principle can then be explicitly represented in a formal manner. Note that the action knowledge of a known solution principle merely involves the changes that are directly related to its function. As to the other changes that may come from indirect influences or side effects, they are not included in the above functional knowledge representation model, and, therefore, are not elaborated here.

4.3 A functional knowledge representation schema

Based on the above research, a schema for representing the function of a known solution principle can then be proposed, as shown in the upper part of Fig. 2. In this schema, the objects in the *Given_Func_Obj* list are those functional objects that need to interact with a solution principle, while the objects in *PS_Obj* list are the enabling objects that a solution principle must have to achieve the desired functionality. For example, in order for the water dissolving effect-based separating solution to work, it must have the enabling object *water*; otherwise, the solution principle would not be able to work.

Note that the whole functional representation rule of a known solution principle has been decomposed into a set of basic functional rules in the above schema, so that it is convenient to implement the functional representation approach in a CACD system. In a basic functional rule, an attribute or a relation in the precondition representation can merely have one value, which is slightly different from that proposed in the above precondition representation model. Therefore, if the precondition of a whole functional rule involves some attributes or relations that have multiple possible values, the precondition should be exhaustively transformed into multiple independent preconditions for indexing multiple basic action rules.

A functional knowledge representation case is also given in Fig. 2, which shows how to represent the functional knowledge of a water dissolving effect-based solution for separating two solid objects. Note that the first rule of the case means that when the input object $Obj1_{in}$ has much better water-solubility than $Obj2_{in}$, i.e., $water\_solubility\langle Obj1_{in}, Obj2_{in}\rangle$ = ">>", what the functional changes will be. There should also be another rule showing the contrary situation, i.e., the input object $Obj_{in}1$ has much worse water-solubility than $Obj_{in}2$, which is not shown in the figure for conciseness. In addition, rule 2 in Fig. 2 is one regarding a situation similar to rule 1, which is represented with a detailed combination to show the water-solubility difference between the input objects.

[Fig. 2. A rule-based functional knowledge representation schema and an example]

It can be found that the proposed functional representation approach can explicitly represent the functional knowledge of a known solution principle. Note that the proposed functional representation approach can also be used to represent a flow or state transformation-focused solution principle, where the redundant flows (i.e., $Obj2_{in}$ and $Obj2_{out}$) and relations in the schema can be neglected. It can be found that our functional representation approach has three major advantages. Firstly, it is independent of any specific material flows, which makes a known solution principle more easily to be reused. Secondly, the precondition representation model allows design experts to externalize the implicit requirements (e.g., disposition requirements) of a known solution principle on the input objects, which a CACD system can then use to rule out the infeasible solution principles for a desired function. Finally, since the action representation model can formally represent the state or relation changes, which often exist as implicit knowledge behind the verb-noun pair representation, it is then possible that a CACD system can use such state or relation changes to achieve functional reasoning.

5 A simulation-based approach for retrieving solution principles

It can be found that the schema for representing a desired MFP function is largely different from the rule-based schema for representing the functional knowledge schema of a known solution principle. Therefore, it is impossible to employ a simple search approach (e.g., the keywords-based search) to retrieving suitable solution principles for a desired MFP function. Therefore, this section will develop a simulation-based approach for achieving the solution principle-retrieving task.

The simulation-based approach is composed of three major processes. Firstly, the CACD system selects a suitable solution principle that can act on the input objects and their relations. Secondly, the CACD system employs the action knowledge of the solution principle to simulate the action on the input objects and their relations, with a result of some output objects and their relations. Finally, the CACD system judges whether the output objects can satisfy the requirements of the desired function or not, so that suitable solution principles that can achieve the desired function can be retrieved. Based on the above idea, a simulation-based algorithm for retrieving feasible solution principles for a desired MFP function can then be proposed. A brief introduction to the simulation-based retrieval algorithm is given as below:

Stage 1: A designer inputs into the CACD system a desired MFP function, which involves the attribute-based representation of input objects, the relation-based representation, and the requirements on the output objects and their relations.

Stage 2: For each solution principle in the knowledge base, the CACD system checks each of its basic functional rules to see whether its precondition (i.e. the requirements on the input objects and their relations) can be satisfied by the current input objects of the desired function; if true, set the solution as the current solution, and go to next step; otherwise, continue to check the next solution, until all solution principles have been searched.

Stage 3: According to the action knowledge of the current solution, the CACD system changes the states of the input objects and the relations between them, with a result of some output objects and some new relations between these output objects.

Stage 4: The CACD system judges whether the output objects and the relations between them can satisfy the requirements of the desired function on the output objects and their relations; if true, add the current solution to the suitable solution principle list; return to Stage 2 to check the next solution.

Stage 5: If all solution principles in the knowledge base have been searched, the CACD system displays all suitable solution principles stored in the solution principle list to designers, from which designers can then select some as promising solution principles for further exploration; if no suitable solution principles retrieved, exit with failure.

Stage 6: Exit with success.

Note that the above simulation-based search process is primarily developed for assisting designers in retrieving suitable solution principles for desired functions. This process is not for achieving automated solution principle synthesis, i.e., it cannot combine multiple solution principles together for achieving a desired function. A primary reason consists in that the material flows often have many disposition attributes, resulting in that it is difficult to predict all unintended changes (including side effects) when a solution principle acts on some new material flows. A possible solution to addressing this issue is to develop an interaction prediction system, which requires additional research that can be carried out in the future.

6 Test

With the aid of a SQL database and a web application development tool, the functional representation methodology and the simulation-based solution principle-retrieving approach proposed before have been implemented as a CACD prototype system for test. The prototype system primarily has three subsystems, i.e., the basic data management subsystem, which is for knowledge engineers to manage some basic data (e.g., object classes, attributes, values), the solution principle knowledge modeling subsystem, which is for domain experts to model the design knowledge of a known solution principle (e.g., the basic information, the functional knowledge, the performance knowledge), and the solution principle-retrieving subsystem, which allows a designer to input a desired function and can retrieve suitable solution principles for it. Hereby, how to employ the prototype system to retrieve suitable solution principles for a desired function, to sort coins, is illustrated as below to demonstrate the proposed research.

6.1 The representation of the desired function

With the prevalence of conductor-free bus services, transportation service companies often obtain a huge number of coins every day. There is an urgent need for developing an efficient coins-sorting device that can rapidly sort different kinds of coins. In this case, the device is assumed to separate two kinds of Chinese coins, i.e., dollar coins (i.e., one-dollar coins) and dimes (i.e., ten-cent coins). Note that dollar coins are primarily made from nickel, while dimes are made from aluminum.

To enable the prototype system to work, a designer should first represent the desired function with the formal functional representation approach proposed in Section 2. Here, the desired function is to sort coins, i.e., to separate dollar coins from dimes. It is evident that this function is a relation transformation-focused function, where the related objects are dollar coins and dimes. The designer should input into the prototype system the attribute-value representation of the two kinds of coins, the concerned relation change, and the requirements on the output coins. Three of the software modules for inputting a desired function are as shown in Fig. 3. Here, the concerned input relation is that the two kinds of input coins are at the same location, i.e., *location (Dollar-Coins, Dimes) ="="*, while the desired output relation is that the two kinds of coins should not be at the same location, i.e., *location (Dollar-Coins, Dimes)* = "≠". In addition, there are also some additional requirements on the output coins, e.g., *Dollar-Coins.state-of-matter* = "SOLID", *Dimes.state-of-matter* = "SOLID".

[Fig. 3. Some user interfaces for inputting a desired function]

6.2 The solution principle knowledge base

The known solution principles in this research primarily come from the previous knowledge base developed by Feng et al. (1996) and Chen et al. (2005), which contained more than 500 solution principles for achieving various MFP functions. Due to limited time, it is impossible to input all these solution principles into the new knowledge base developed in this research. In this test, six related solution principles are chosen from the previous knowledge base to demonstrate the solution principle-retrieving approach, with their sketches shown in Fig. 4.

[Fig. 4. Some sketches of the known solution principles]

The first solution principle (sketch a), called the dissolving effect-based solution, comes from Asian washing machines. Through dissolving dirt into detergent and water, it separates dirt from clothes. The second solution principle (sketch b), called the sieving effect-based solution, comes from a materials-sieving device. Based on the vibration of the sieve, smaller materials fall down, while bigger materials remain on the sieve. The third solution principle (sketch c), called the high electrical voltage effect-based solution, employs the strong electrical field generated by high voltage to separate the particulates with good electrical-conductibility from those with poor electro-conductibility. The fourth solution principle (sketch d), called the electro-magnetism effect-based solution, could separate particulates that can be magnetized from those unable to be magnetized. The fifth solution principle (sketch e), called the air-blowing effect-based solution, comes from a device for separating beans from shells. It employs the airflow generated by an electrical fan to separate materials with bigger density from those with smaller density. The last solution principle (sketch f), called the melting effect-based solution, can separate materials with a lower melting point from those with a higher melting point.

Fig. 5 shows three primary modules of our prototype system for managing solution principle knowledge. One is for domain experts to input the basic information of a known solution principle (e.g., solution type, solution name, functional objects), another is for managing the state or disposition precondition for a known solution principle, and the other is for managing the relation precondition of a known solution principle. There are also some other modules for managing the functional knowledge about a solution principle, which, due to limited space, are omitted here.

[Fig. 5. Some interfaces for managing the functional knowledge of a solution principle]

6.3 The solution principle-retrieving process

Given the representation of the desired function and the solution principle knowledge base, the prototype system can then employ the simulation-based approach to retrieve suitable solution principles for the desired function of sorting coins. To illustrate the solution principle-retrieving process, it is assumed that the solution principle knowledge base here merely includes the six solution principles mentioned above.

The CACD system will first analyze the first solution principle, i.e., the dissolving effect-based solution, whose functional knowledge representation is similar to that shown in Fig. 2. Since the precondition of this solution principle requires that one input object should have much better water- or detergent-solubility than the other, which cannot be met by the two kinds of coins, the CACD system can then rule out this solution, and continue to analyze the next solution.

The second solution principle is the sieving effect-based solution, which requires that the input objects should be with obvious size difference, i.e., $size(Obj1, Obj2) \in \{\text{">"}, \text{">>"}\}$ or $size(Obj1, Obj2) \in \{\text{"<"}, \text{"<<"}\}$. This precondition can be met by the two kinds of coins, since there is an obvious difference between them, i.e., $size(Dollar\text{-}Coins, Dimes) = \text{">"}$. Therefore, the prototype system can then know that this solution principle is suitable for achieving the coins-sorting function. The CACD system can then employ the action knowledge of the solution principle to change the input objects and their relations, resulting in that the location relation between the output coins is set as: $location(Dollar\text{-}Coins, Dimes) = \text{"}\neq\text{"}$, which can meet the relation requirement of the coins-sorting function on the output coins. Therefore, this solution is retrieved as a suitable solution for the coins-sorting function.

The prototype system will continue to analyze other solution principles. Among the six solution principles, the prototype system will finally retrieve three suitable solution principles for achieving the coins-sorting function, i.e., the sieving effect-based solution, the electro-magnetism effect-based solution, and the air-blowing effect-based solution. Fig. 6 shows a user interface that displays the desired function and the retrieved solution principles, and a user interface that displays the detailed functional knowledge of the retrieved solution principle, i.e., the electro-magnetism effect based solution.

[Fig. 6. Some interfaces for displaying the solution principle-retrieving results]

Note that the sixth solution principle, i.e., the melting effect-based solution, cannot be retrieved as a suitable solution, although the melting-point difference between the coins can meet the precondition required by the solution principle, i.e., $melting\_point(Obj1, Obj2) \in \{">", ">>"\}$ or $melting\_point(Obj1, Obj2) \in \{"<", "<<"\}$. This is because this solution principle has a function-related melting action that can change the state-of-matter of coins with lower melting point into liquid, and thus cannot meet the requirement on the output coins, i.e., *Object1.state-of-matter* = "SOLID" & *Object2.state-of-matter* = "SOLID".

6.4 Discussion

It can be found that after a designer inputs a desired function into the prototype system, the prototype system can then retrieve suitable solution principles from the knowledge base consisting of solution principles from various domains (e.g., mechanical, electrical, electromagnetic), and then outputs the solution principles for the designer. Therefore, based on the proposed functional representation methodology and the simulation-based search approach, the prototype system can effectively assist designers in explicitly representing a MFP function and in retrieving suitable solution principles for a desired MFP function. Note that the proposed functional representation methodology can also be used for conceptual design of energy or signal flows-processing systems, since the proposed methodology is also suitable for representing the functions dealing with energy flows and signal flows. In addition, it can also be found that the CACD system can alleviate designers' biases towards familiar solution principles, since it can assist designers in retrieving all possible solution principles without any biases.

However, it should be admitted that the proposed solution-retrieving approach also requires much more functional information (i.e., the material flow representation) than the traditional approaches, which could be a heavy burden for designers. A possible solution to alleviate this burden is to provide designers with additional knowledge base of material flows, so that designers can easily reuse such knowledge when defining a desired function.

7 Conclusions

Due to the significance of conceptual design, much CACD research has been carried out in recent decades. Although many domain-specific CACD approaches have been developed for conceptual design of energy flows-processing systems, there is little research for conceptual design of MFP (i.e., Material Flows-Processing) devices. Since the solutions of MFP functions often come from various domains, it is then impossible to employ the traditional domain-specific approaches to achieve the conceptual design of MFP devices. Therefore, this research has been devoted to developing a general (i.e., domain-independent) and knowledge-based methodology for achieving the conceptual design of MFP devices.

The primary contributions of this research are as follows. Firstly, it develops a flexible ontology-based approach to representing material flows, and their relations and changes, so that a desired MFP function can be represented in an unambiguous manner. Secondly, it proposes a rule-based approach to representing the functional knowledge of a solution principle, so that the functional knowledge can be represented in a general and flexible manner. Finally, it develops a simulation-based approach for retrieving suitable solution principles for desired MFP functions, which can select suitable solution principles from the knowledge base to act on the input material flows and their relations to generate output(s). As seen in the test, the proposed approaches allow the CACD prototype system to effectively and precisely retrieve suitable solution principles for a desired MFP function.

This research has dealt with how to represent MFP functions and how to retrieve solution principles for them, without additional requirements taken into consideration. Since more requirements can be gathered as designing proceeds, which often results in the search in an identified solution space for viable solutions, a more flexible design search approach should be developed in the future. Another future work can be to develop an automated functional reasoning approach, so that a CACD system can automatically combine the known solution principles into a combinatorial solution to address a MFP function that does not have a solution principle in the knowledge base. In addition, since conceptual design is a complex process that involves many other kinds of tasks, another future work can be to integrate our solution-searching approach with other CACD approaches (e.g., functional modeling approaches, systems validation approaches) to provide more complete support for conceptual design engineers.

Acknowledgements

This research is supported by the Natural Science Foundation of China (No. 51375301, 51575342) and by the State Key Laboratory of Mechanical System and Vibration at Shanghai Jiao Tong University (No. MSVZD201401).

References

Arnold, C. R. B., Stone, R. B., & McAdams, D. A. (2008). MEMIC: an interactive morphological matrix tool for automated concept generation. *Proc. Int. Industrial Engineering Annual Conf.* Norcross: Institute of Industrial Engineers-Publisher.

Bunge, M. (1977). *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*. Dordrecht: D. Reidel Publishing Company.

Cagan, J., Campbell, M. I., Finger, S., & Tomiyama, T. (2005). A framework for computational design synthesis. *ASME Journal of Computing and Information Science in Engineering 5(3)*, 171-181.

Campbell, M. I., Cagan, J., & Kotovsky, K. (2000). Agent-based synthesis of electromechanical design configurations. *ASME Journal of Mechanical Design 122(1),* 61-69.

Chakrabarti, A., & Bligh, T. P. (1996). An approach to functional synthesis of mechanical design concepts: theory, applications, and emerging research issues. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10(4)*, 313-331.

Chandrasekaran, B. (1994). Functional representation: A brief historical perspective. *Applied Artificial Intelligence 8(2)*, 173-197.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems 14(1)*, 20-26.

Chen, Y., Lin, Z., & Feng, P. (2005). A genetics-based approach for the principle conceptual design of mechanical products. *The International Journal of Advanced*

Manufacturing Technology 27(3-4), 225–233.

Chen, Y., Feng, P., He, B., Xie, Y., & Lin, Z. (2006). Automated conceptual design of mechanisms based on improved morphological matrix. *ASME Journal of Mechanical Design, 128(3)*, 516-526.

Chen, Y., Zhang, Z., Huang, J., & Xie, Y. (2013). Towards a Scientific Ontology-based concept of function. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 27(3)*, 241-248.

Chen, Y., Zhang, Z., Xie, Y., & Zhao, M. (2015). A new model of conceptual design based on Scientific Ontology and intentionality theory. Part I: The conceptual foundation. *Design Studies 37*, 12-36.

Deng, Y. M., Britton, G. A., & Tor, S. B. (2000). Constraint-based functional design verification for conceptual design. *Computer-Aided Design 32(14)*, 889-899.

Eisenbart, B., Gericke, K., & Blessing, L. (2013). An analysis of functional modeling approaches across disciplines. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 27(3)*, 281-289.

Feng, P., Xu, G., & Zhang, M. (1996). Feature modeling based on design catalogues for principle conceptual design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10(4)*, 347–354.

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

Goel, A. K. (2013). A 30-year case study and 15 principles: implications of an artificial intelligence methodology for functional modelling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 27(3)*, 203-215.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies 43(5)*, 907-928.

Gu, C. C., Hu, J., Peng, Y. H., & Li, S. (2012). FCBS model for functional knowledge representation in conceptual design. *Journal of Engineering Design 23(8)*, 577-596.

Helms, B., Schultheiss, H., & Shea, K. (2013). Automated mapping of physical effects to functions using abstraction ports based on bond graphs. *ASME Journal of Mechanical Design 135(5)*, 051006.

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., & Wood, K. L. (2002). A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design 13(2)*, 65-82.

Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo: Morgan Kaufmann.

Kuipers, B. (1986). Qualitative Simulation. *Artificial Intelligence 29(3)*, 289–338.

Kurtoglu, T., Campbell, M. I., Arnorld, C. B., Stone, R. B., & McAdams, D. A. (2009). A component taxonomy as a framework for computational design synthesis. *ASME Journal of Computing and Information Sciences in Engineering 9(1)*, 011007.

Kurtoglu, T., Swantner, A., & Campbell, M. I. (2010). Automating the Conceptual Design Process: From Black Box to Component Selection. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 24(1)*, 49–62.

Lenat, D. B. (1989). Ontological versus knowledge engineering. *IEEE Transactions on Knowledge and Data Engineering 1(1)*, 84-88.

Li S., Hu J., & Peng Y. H. (2010). Representation of functional micro-knowledge cell (FMKC) for conceptual design. *Engineering Applications of Artificial Intelligence 23(4)*, 569-585.

Miles, L. D. (1972). *Techniques of Value Analysis and Engineering*. New York: McGraw-Hill.

Pahl, G., & Beitz, W. (2007). *Engineering Design: A Systematic Approach* (3rd Edition). London: Springer-Verlag.

Prabhakar, S., & Goel, A. K. (1998). Functional modeling for enabling adaptive design of devices for new environments. *Artificial Intelligence in Engineering 12(4)*, 417-444.

Rosenberg, R., & Karnopp, D. (1983). Introduction to Physical Systems Dynamics. McGraw-Hills, New York.

Schmidt, L. C., & Cagan, J. (1997). GGREADA: a graph grammar-based machine design algorithm. *Research in Engineering Design 9(4)*, 195-213.

Qian, L., & Gero, J. S. (1996). Function–behavior–structure paths and their role in analogy-based design. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing 10(4)*, 289-312.

Sen, C., Summers, J. D., & Mocko, G. M. (2013). Physics-based reasoning in conceptual design using a formal representation of functional structure graphs. *ASME Journal of Computing and Information Science in Engineering 13(1)*, 011008.

Sridharan, P., & Campbell, M. I. (2005). A study on the grammatical construction of function structures. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing 19(03)*, 139-160.

Stone, R. B., & Wood, K. L. (2000). Development of a functional basis for design. *ASME Journal of Mechanical Design 122(4)*, 359-370.

Stone, R. B., Tumer, I. Y., & Stock, M. E. (2005). Linking Product Functionality to Historic Failures to Improve Failure Analysis in Design. *Research in Engineering Design 16(2)*, 96–108.

Sturges, R. H., O'Shaughnessy, K., & Kilani, M. I. (1996). Computational model for conceptual design based on extended function logic. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing 10(4)*, 255-274.

Szykman, S., Racz, J., Bochenek, C., & Sriram, R. D. (2000). A web-based system for design artefact modelling. *Design Studies 21*, 145-165.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function–behavior–state modeller. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing 10(4)*, 275–288.

Ulrich, K. T., & Seering, W. P. (1989). Synthesis of schematic descriptions in mechanical design. *Research in Engineering Design 1(1)*, 3–18.

Welch, R. V., & Dixon, J. R. (1994). Guiding conceptual design through behavioural reasoning. *Research in Engineering Design 6(3)*, 169-188.

Wölkl, S., & Shea, K. (2009). A computational product model for conceptual design using SysML. *Proc. ASME 2009 Int. Design Engineering Technical Conf. & Computers and Information in Engineering Conf.*, Paper No. DETC2009-87239, San Diego, California, August 30-September 2.

Figure List

Author biographies

Yong Chen is an Associate Professor of engineering design in the School of Mechanical Engineering at Shanghai Jiao Tong University, China. He received his bachelor and PhD degrees from Zhejiang University, China. He joined Shanghai Jiao Tong University as a postdoc in 2004. He was a visiting scholar in the IMPACT Laboratory at University of Southern California for 1 year. Dr. Chen's research interests include many aspects of engineering design research, in particular conceptual design, design innovation, design knowledge reuse, and computer-aided design.

Meng Zhao is currently a PhD student in the School of Mechanical Engineering at Shanghai Jiao Tong University, China. He received his bachelor degree in 2009 from Zhejiang University, China. His research interests include conceptual design, computer-aided design, and systems engineering.

Ying Liu is a Senior Lecturer in the Institute of Mechanical and Manufacturing Engineering at the School of Engineering, Cardiff University, UK. He obtained his Bachelor and Master both in Mechanical Engineering from Chongqing University, China in 1998 and 2001, and Ph.D. from National University of Singapore in 2006. He was an assistant professor with Hong Kong Polytechnic University and National University of Singapore. He joined Cardiff University in 2013. His research interests include design theory and methodology, design informatics, data mining and machine learning.

Youbai Xie is a Professor in the School of Mechanical Engineering at Shanghai Jiao Tong University, China. He is the Director of the Institute of Modern Design of Shanghai Jiao Tong University. Dr. Xie is an academician of the Chinese Academy of Engineering. His research interests include engineering tribology, design science, and knowledge management.