# ANALOGICAL MATCHING USING DEVICE-CENTRIC AND ENVIRONMENT-CENTRIC REPRESENTATIONS OF FUNCTION

GREG P MILETTE AND DAVID C BROWN
*Worcester Polytechnic Institute, USA*

**Abstract.** This research experiments with representations of function using analogical matching, trying to determine the benefits of using environment-centric (EC) vs. device-centric (DC) representations. We use the Structure Mapping Engine for matching, and seek to show the effect on quality and quantity of analogical matches when the representation is varied.

## 1. Introduction

Designing something is challenging, so providing computational help is important. Software systems can help the designer, or might replace the designer in some situations (Brown 1992). For computers to design devices we have to describe them in some way using a knowledge representation.

This research experiments with two different knowledge representations: both based on the Structure-Behavior-Function model for describing devices (Chandrasekaran and Josephson 2000). That work describes two different ways to represent the function of devices, Device-Centric (DC) and Environment-Centric (EC), both described below. Each may be advantageous in certain situations, but there appears to be no research showing what the effects are of using DC vs. EC representations of function.

This research seeks to perform experiments that will explore what the effects are with certain design tasks. The knowledge representation will be used in experiments with some automated reasoning, producing results that we can measure.

Motivated by exploring computational support for creativity (Boden 1994), we target analogical reasoning. Analogy is often cited as a key ingredient of creativity (Goel 1997; Gentner et al. 2001). In addition, functional reasoning is also at the right level of abstraction to support creativity (Umeda and Tomiyama 1997).

Analogical reasoning involves expressing what the current situation is, looking for past situations that might apply (matching), and finally applying



*J.S. Gero (ed.), Design Computing and Cognition '06*, 183–202.

them to the current situation (transfer). A full study would require a system that performs all the steps in analogical reasoning, but for this research we take the first step and focus only on the matching phase. We use an algorithm called SME (Falkenhainer et al. 1989). SME was chosen because it is well tested in much research, it is claimed to have psychological backing, the software is available, and because it is suited for the problem.

Using SME we can take a pair of devices represented with a particular knowledge representation and produce a list of possible matches between them with associated weights. We measure the quantity and quality of the matches in order to measure the effect of DC vs. EC representations.

We are also interested to see whether novel matches are produced: i.e., whether DC vs. EC representations might have any effect on novelty, a key aspect of creativity (Besemer and Treffinger 1982). Therefore we consult a group of humans to get their judgment.

We have performed a set of experiments that that indicate where the results are coming from: i.e., the credit assignment problem. The issue is whether DC vs. EC representations, or the representation used (level of detail; ontology) should be given credit (Kitamura et al. 2004).

The *hypothesis* of this research is that representations with EC information will produce a greater number of analogical matches, and that these matches will be of lower strength than matches made using representations that only contain DC information. We hypothesize that creating a representation with *both* DC and EC information should produce even more matches than either DC or EC alone, and that these matches should have higher weights.

We show through experimentation with SME that EC produces more matches than DC, DC produces higher quality matches than EC, and, in contrast to our hypothesis, the combined representation produces comparatively fewer matches and more lower quality matches than EC alone.

In addition, from limited experiments with humans we show that they tend to rate low weighted matches as being more novel than high weighted matches and rate DC matches as being more novel than EC matches.

## 2. Design Research

A lot of research has been done on functional representation and reasoning (Umeda and Tomiyama 1997; Stone and Wood 1999; Pahl and Beitz 2003; Chandrasekaran and Josephson 2000; Stone and Chakrabarti 2005.

A full description of a device D's intended function would link it, via relationships and behaviors, to "purpose" Brown and Blessing 2005; Rosenman and Gero 1998). There is little work addressing the effect of a device on the environment. Prabhakar and Goel (1996) distinguish the "external environment" of D, from the "outer environment" of D: i.e., those entities in the environment that directly interact with D.

Our work is most influenced by Chandrasekaran and Josephson's framework (2000). They consider that a device (D) is used by being placed in an environment (E). The causal interactions that result from this "mode of deployment" occur due to a pattern of relationships over time between D and E. If the pattern of behaviors arising from the interactions is *desired* then D performs a function in E. Behaviors are seen as values of, or relations between, state variables, or properties, of an object, considered over time.

If the desired behaviors are expressed in terms of D only, then they consider it to be a Device-Centric (DC) description of D's function. An Environment-Centric (EC) description only uses elements in E: such a description might be presented in the early stages of a design task.

Our representations are a strict DC version, and another that includes all entities that interact, from both D *and* the outer environment. We refer to this, for contrast, as EC, but from Chandrasekaran and Josephson's definitions it should more properly be called "mixed".

### 2.1. USING THE FUNCTIONAL BASIS

The terms used to describe the function of different devices must be consistent and at the same level of abstraction so that device descriptions are comparable. This will reduce the variation and noise in results. For example, using more abstract terms for one device may cause SME to generate more matches, making strong conclusions harder to make, while inconsistent terms may cause fewer matches, with similar consequences.

This research uses a set of domain specific terms called the "functional basis" (Stone and Wood 1991). The functional basis provides a set of domain-dependent terms for flows and functions. The representations in this research use flows in the same way the functional basis does. The functional basis represents flows of material, energy or signal that transfer from one device to the next. The basic functions available include import, export, transmit, couple, display, rotate, and change. Our representation uses the basic functions from the functional basis work as a way of describing device behaviors.

## 3. Knowledge Representation

There are several goals for the knowledge representation (KR):

1. It can represent DC and EC functions;
2. It can represent devices at different levels of detail;
3. DC and EC parts can be combined to form a combined representation. We refer to this representation as BOTH (see example in Section 3.3).

The KR must be descriptive enough to describe functions and must allow for different experiments. These experiments (Section 6) require the ability to represent devices at different levels of detail, and also to use the DC only, EC only, or BOTH versions of the each device's representation.

### 3.1. DESIGN DECISIONS

We consider a function to be a set of desired behaviors. Rather than including all of the constructs from Chandrasekaran and Josephson's work, such as mode of deployment, this research represents only behaviors and functions, leaving further exploration of Chandrasekaran and Josephson's concepts to future work.

The KR is somewhat independent from SME vocabulary, but is still easily translatable into proper input for SME. This decouples the KR from the particular intricacies of the matching algorithm implementation used.

### 3.2. OBJECTS IN THE REPRESENTATION

There are five main concepts in the KR: devices, functions, behaviors, relations, and flows. To completely specify a device using the KR, one must provide a library of relations and flows, a set of behaviors and a set of functions that group the behaviors.

A device has a set of functions that are either DC or EC. Each function consists of a set of behaviors. Since a device may have multiple functions, some of a device's behaviors may be mentioned in more than one function.

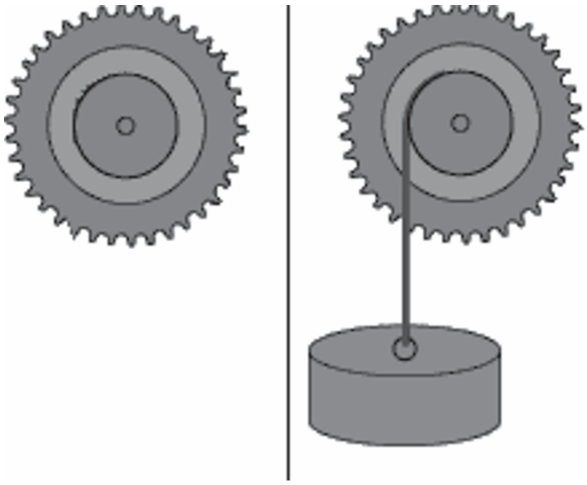
Devices are physical objects in the world and their behaviors describe how they interact. Behaviors are instantiations of relations. The relations (e.g., import) provide constructs that are filled in with domain specific elements, such as flows or other devices, in order to specify a behavior. For example "import <flow> <device>" is an example of a relation with two arguments. Instances are import torque gear and import force drum.

Flows are the material, energy or signals involved in a particular behavior. For example, a behavior change force surface describes how the flow "force" interacts with the device "surface".

The environment for a particular device is an outer environment defined by a set of external objects that interact with the device. It is *not* the entire external environment. The representation does not have an explicit representation of the environment. Instead it describes the environment using behaviors. For example, the behavior transmit torque *minutegear* references *minutegear*, which is part of the environment. Also, the representation can have behaviors that do not refer to the environment at all. To distinguish objects which are part of the environment from the device we mark objects in the environment by underlining them.

### 3.3. USING THE KNOWLEDGE REPRESENTATION

The objects described in Section 3.2 can be used to satisfy the goals we had for the knowledge representation. This section provides examples of devices represented with high and low detail. This section also provides examples of DC and EC behaviors and functions.



(a) (b)

*Figure 1.* (a) A gear (b) a gear and a weight. Other devices that interact with these two, earth and gear2, are not shown.

The KR can be used to represent DC behaviors and functions for the gear pictured in Figure 1(a)s. The relation “import <flow> <device>” is used to define the behavior:

import force gear (b1)

The relation “export <flow> <device>” is used to describe the result of behavior b1:

export force gear (b2)

The two behaviors combine to form a single DC function.

b1, b2 (dc1)

To represent EC behaviors and functions, the representation needs to introduce another device to interact with the gear because EC behaviors need to mention something in the environment.

For the situation with a weight and two gears, partially represented in Figure 1b, two EC behaviors are available for the gear:

transmit force from *weight* to gear (b3)

transmit force from gear to *gear2* (b4)

The environment of the gear consists of *weight* and *gear2*. The behaviors b3 and b4 combine to form an EC function:

b3, b4 (ec1)

The weight in the mechanism can also be represented with two behaviors:

transmit force from *earth* to weight (b5)
transmit force from weight to *gear* (b6)

The environment of the weight consists of *earth* and *gear*. The behaviors b5 and b6 combine to form an EC function for the weight:

b5, b6 (ec2)

When representing with low detail, the representation focuses on a particular device. The device has no internal components and the behaviors for the device either refer to the device itself or to objects in the environment.

For a high detail representation, the KR needs to combine low detail descriptions together. The representation does this by combining the behaviors and functions from each low detail device. Using the gear and weight example from figure 1b the high detail EC function would contain four behaviors instead of two and only one function. The EC function would be:

b3, b4, b5, b6 (ec3)

This KR can be used to create a BOTH representation by concatenating the EC and DC version of each device representation. Thus, the BOTH representation for the gear consists of the functions dc1 and ec1 as well as the behaviors b1, b2, b3, and b4. Note that as the DC and EC representations use different relations there is no overlap when constructing the BOTH representation.

## 4. SME

This section briefly describes how SME works and what features are relevant for this research (Falkenhainer et al. 1989). The code used in this research is available online (Falkenhainer 2005).

The SME algorithm takes two devices, called the source and target, and maps knowledge from the source into the target. The first step of the algorithm is to create a set of match hypotheses. A match hypothesis represents a possible mapping between a part of the source and the target.

SME uses match rules to calculate positive and negative evidence for each match. SME combines different amounts of evidence together, favoring matches between parts of the device representation that have similar relation names.

For example, given relations r1 and r2:

transmit torque inputgear secondgear (r1)
transmit signal switch div10 (r2)

SME produces four match hypotheses. Three of the match hypotheses are torque to *signal*, *inputgear* to *switch*, and *secondgear* to *div10*. Each has a positive evidence value of 0.6320. SME also matches both "transmit" relations and gives them 0.7900 positive evidence. These matches are between flows, between devices, and between relations. All of these matches are dependent on the shared relation name "transmit".

The match rules also propagate evidence from higher matches down to lower matches. This gives additional evidence to matches that are part of a higher order relation match. SME does this because of the Systematicity Principle which states that more connected knowledge is preferred over independent facts (Falkenhainer et al. 1989).

SME can produce negative evidence when the relation type matches, but the elements in the relation do not match.

The rest of the SME algorithm is involved in creating maximally consistent sets of match hypotheses. These sets are called "gmaps". The sum of all the positive evidence values in the match hypotheses of a gmap becomes the weight of the gmap. Comparing a source device to a target device may produce one or more of these gmaps, each with an associated weight. Also, SME combines multiple smaller gmaps to produce bigger, maximally consistent, gmaps.

### 4.1. SME PROPERTIES RELEVANT TO THIS RESEARCH

Since this research is comparing two kinds of KR, there are some properties of SME that are relevant for the determining reasons why one KR produces different results than another:

- More information in a particular representation should allow for matches of higher weight. This is because longer representations can produce more match hypotheses and thus have higher weighted gmaps.
- Making longer representations may not produce a greater number of gmaps because gmaps can be combined together during the creation of maximally consistent gmaps

Our experiments use these properties to explain why results using the DC and EC representations differ.

## 5. Test Examples

The requirements for the test examples to be used are that they: must have varied levels of detail; must include both DC and EC representations; should be similar enough to allow analogical matches; should allow for novel matches; must be a large enough sample so that general conclusions can be reached; and must be capable of being understood by humans.

The test examples used in this research are a set of clocks that are decomposed into components and subcomponents. By combining different subcomponents together, the level of detail can be adjusted. Because the clocks share components, there are obvious analogical matches that SME can make, providing good contrast for results that people may consider novel. The test examples represent 21 individual subcomponents, which can be grouped into 8 larger components.

### 5.1. THE CLOCK TEST EXAMPLES

We use two kinds of clocks: a digital clock, such as a bedroom alarm clock, and a pendulum clock, such as a grandfather clock. Each clock has a different way to achieve the functions of setting and displaying the time.

Each clock works differently, but they share common components and common functions. These components are the powerprovider, which provides some kind of energy into the clock, the timebase, which converts the energy into a periodic signal, a gear, which converts the signal into a once-per-second or once-per-minute signal, and a face which displays the time.

We used articles by Brain (2005a; 2005b) as sources of information about clocks. When using a clock a human needs to observe the time and be able to set the time. Figure 2 shows a conceptual diagram of these components and how they interact. Arrows indicate the direction of flow in the clock. For example, the powerprovider transfers energy to the timebase. The human interacts with the clock by resetting it or by receiving a visual signal.

Figure 3 shows a schematic for a pendulum clock. The schematic labels all the pendulum clock's components. Figure 4 shows how these subcomponents get grouped into components. For example, the secondhand and minutehand are subcomponents of face. The schematic for the pendulum clock is shown in Figure 5.

The pendulum clock works primarily with gears while the digital clock uses many divide-by-x counters. The hierarchy for the digital clock includes subcomponents such as a divide-by-10 counter, which is part of the digital gear, and a plug, which is part of the digital power provider.

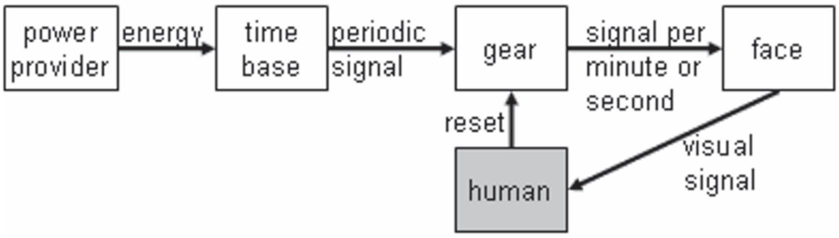



*Figure 2.* Generic model of a clock: components and how they interact with each other and with a human.

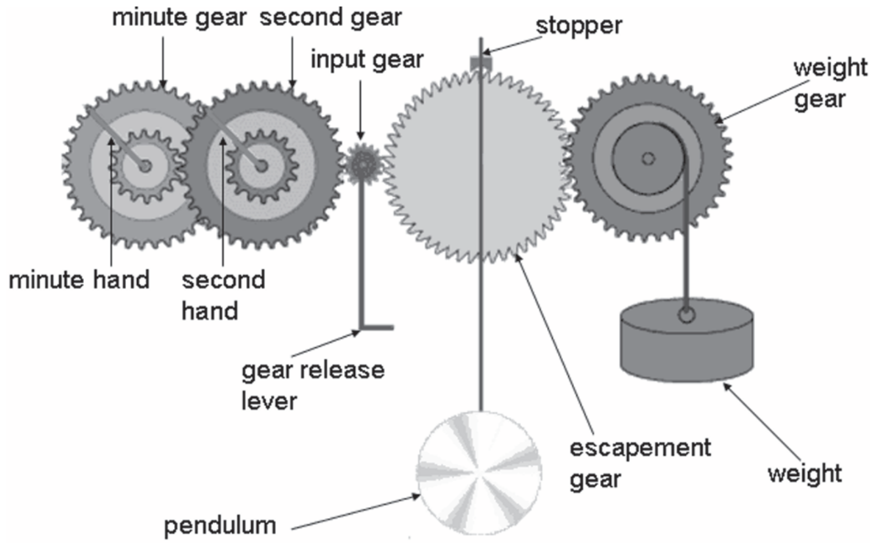



*Figure 3.* Schematic for an idealized pendulum clock showing all its components. Diagram based on (Brain 2005b).

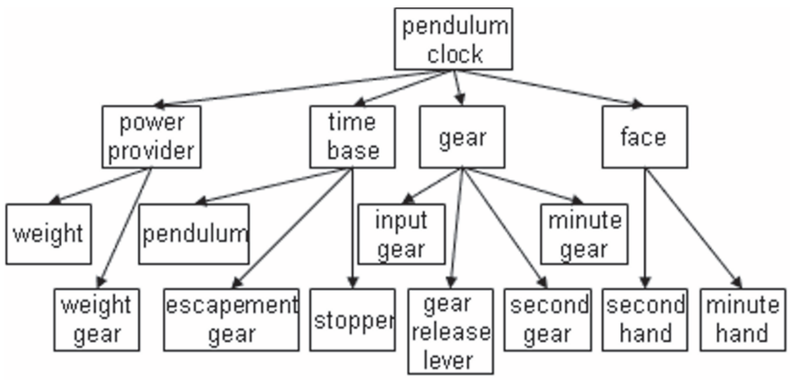



*Figure 4.* Hierarchy for the pendulum clock. Boxes show the devices; arrows represent a component-subcomponent grouping.

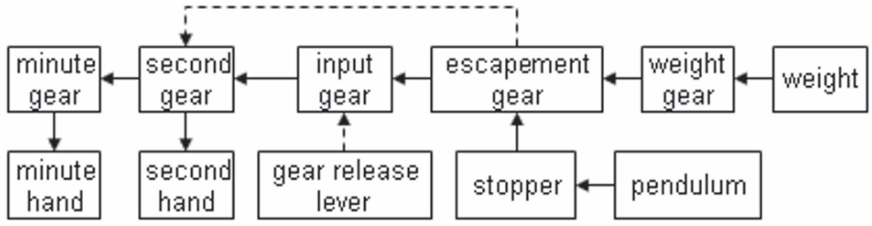



*Figure 5.* Schematic for pendulum clock. Boxes represent subcomponents; solid arrows represent flow; the dotted line represents flow when the gear release lever is pressed.

Thus, the test examples are made up of two different clocks that can be represented at two levels of detail. The low detail representations are the subcomponents of the clocks such as *secondgear* or *plug*. The high detail representations include clock components such as digital *powerprovider*.

## 6. Computational Experiment

SME produces a list of gmaps for each match, each with an associated weight. The goal of the computational experiment is to analyze these lists of gmaps and explain how they are affected by different representation types. Overall, the experiment demonstrates the following effects:

- EC has lower weighted matches than DC;
- EC generates more matches than DC;
- EC matches have higher variance than DC;
- BOTH matches are fewer in number and have lower weights than DC or EC alone.

The experiment and analysis must be able to measure these effects, explain them, and show that they are robust. The experiment measures the gmap weights, the gmap weight variance, and the number of gmaps generated. To make fair comparisons between the datasets the gmap weights and number of gmaps are normalized.

The experimental results can be influenced by several factors including the representation length, the representation complexity, and the number of devices mentioned. Each experiment is run on low and high detail test examples in order to show that any observed effects remain the same even when the level of detail is varied.

### 6.1. EXPERIMENTAL RUNS

The experiment uses the factorial experiment design shown in Table 1. Overall, there are 6 different device sets that the experiment uses. There are three versions of device representations, EC, DC, and BOTH. Each version is categorized into low detail and high detail. Each combination makes up an experiment test set. There are 21 low detail and 8 high detail devices. The devices are the 21 subcomponents and 8 components of the clocks described in Section 5.

TABLE 1. Factorial experiment design showing the 6 different device sets.

| | **Low detail** | **High detail** |
|---|---|---|
| **EC** | | |
| **DC** | | |
| **BOTH** | | |

An experiment test run consists of analyzing pairs of devices from a particular test set. SME compares each device in the test set to the other devices in the test set. The experiment disregards comparisons between the same device. This results in $n^2–n$ comparisons where n is the number of devices in the test set. For example, the low detail test set has 420 matches in it.

### 6.2. EXPERIMENTAL FACTORS

This experiment needs to show how the gmap weight, gmap weight variance, and number of gmaps differ for the EC, DC, and BOTH datasets. This is complicated by the fact that several factors can affect these statistics.

The representation length is the sum of the number of functions and behaviors in the source representation. We find that for most of the data, the representation length and the number of gmaps are positively correlated ($p<0.05$). This means that as the representation length increases more gmaps get generated. Our normalization procedure decreases this correlation.

The representation complexity is the sum of the number of behaviors in each function and the number of arguments in each behavior divided by the representation length. For example, the DC version of the gear from Section 3.3, with behaviors b1, b2 and function dc1, has a representation complexity of 2. This measure of complexity is similar to the one used in (Balazs 1999).

In our data, on average, EC representations have the highest amounts of complexity. This is because DC representations only mention the device, and EC representations mention both the device and the environment.

### 6.3. NORMALIZED GMAP WEIGHT AND VARIANCE

The experiment needs to compare the magnitude and variance of the weights between the datasets. The factors described in Section 6.2 imply that the gmap weights cannot be compared directly unless some aspects of the representation are taken into account.

Therefore, we use a normalization strategy in order to make a fair comparison between the representations. The normalization formula first computes the value, MAXVAL, which is equal to the highest weighted gmap SME produces when the device is compared to itself. Then the weight of each gmap made with that device is divided by MAXVAL to obtain a new normalized weight.

This strategy should adjust the magnitudes of the gmap weights to account for both the representation length and complexity. It also gives the measurement more meaning. Instead of measuring its overall strength, this normalized weight measures the relative amount of a device's representation that is matched by the target device. Thus, the higher the normalized weight, the more of the target device fits with the source device.

Each time SME generates a match it outputs a list of gmaps, each of which has an associated weight. Since our comparisons are done on a per match basis and not on a per gmap basis we need to aggregate the gmap weights for each match and then use the aggregated result for our analysis. Thus, for each match, we compute the average, standard deviation, and highest of its gmap weights. Then, for all matches we compute additional statistics to create results such as "average of average gmap weights" or "average standard deviation of gmaps."

### 6.4. NORMALIZED NUMBER OF GMAPS

The number of gmaps is positively correlated with the representation length. In order to account for this influence and to compare the different datasets, we normalize the data by the representation length. Unlike the gmap weight measure, we could not use the number of gmaps generated when a device is compared to itself because it was not close to an upper bound on the number of gmaps.

The formula for computing the normalized number of gmaps is the number of gmaps divided by the representation length. For example, if a match has a representation length of 5 and generates 10 gmaps, then the normalized number of gmaps would be 2.

## 7. Human Experiment

In the computational experiments we present SME with representations of two devices, and it outputs a list of potential matches between portions of each representation. For example, based on these lower level matches, SME might suggest that a pen is like a hammer.

As our hypothesis concerns the possible benefits of different styles of device representation, the representation is varied throughout the experiments, and the resulting matches are measured and evaluated.

We are interested in performing the human experiment for two reasons. First, we would like to determine whether or not the matches proposed by SME are “novel”: e.g., a pen is like a sponge. We hypothesize that one form of device representation is less likely to produce novel matches.

Second, we would like to investigate how the match weights generated by respondents correlate with SME match weights. There are two ways the SME results can correlate.

First, the human and SME results could place the same relative weights on certain device matches. For example, both the human and SME could think that the pen is more like a hammer than it is like a sponge.

Second, the human results can lend support to the DC or EC representation if the reasons the humans are using match with the representation that SME uses, and if the human's and SME’s match weights correlate. We might get this result if the human thought the pen was most like a sponge because they both interact with liquid and if SME marked them as most similar because the pen and sponge both interact with a human's hand. Though the reasons are not exactly similar, they both involve EC reasoning: i.e., about how the device interacts with the environment.

To gather this information from human respondents, we use two techniques: repertory grids and a questionnaire (Hart 1986). The respondents are a volunteer group of engineers.

The repertory grid technique provides several benefits:

- It is a proven technique that allows respondents to give information about the similarity of different devices in a group. The result of collecting the grid information is a “percent similar” measure describing the human’s evaluation of device similarity. After normalization, it can be compared to SME output which also reports how similar devices are.
- As part of the grid creation process, respondents give reasons why they differentiated one device from another. This information can be classified as DC or EC, lending support to that approach. It can also be compared directly to the lower level matches in the computer results.
- A good computer tool is available that makes collection of repertory grids relatively easy (Shaw and Gaines 2005).

A questionnaire is used to determine novelty. It asks the respondents to evaluate how novel they think the computer's analogical matches are and to disregard other observations such as correctness when they make their evaluation. The respondents indicate low, medium, or high novelty. The respondents are asked about results from various computer experiments produced using different representations.

The experimental procedure is to first collect a repertory grid and then have the respondent fill out a questionnaire: both about the same devices. Collecting the repertory grid first is important, as it allows the respondents to determine for themselves how the devices relate to each other. Thus, when they fill out the questionnaire, they will be able to compare the computer's answers to their own and be better able to judge the novelty of them.

A subset of the clock examples from the computational experiment is used for the human experiment. Preliminary experiments with simpler examples, such as pens and sponges, indicated that the respondents were using reasons that we could classify as DC or EC. However, many of their reasons were focused on surface features. The clock examples subsequently adopted have similar functions, but very different surface features. Therefore, the respondents tend to focus their attention on the function of the clock components, which is what we want.

Since understanding clocks takes time, the respondents were given articles to read before the experiment. These articles were the same ones that we used to create the representations for the computational experiment, taken from *How Digital Clocks Work* (Brain 2005a) and *How Pendulum Clocks Work* (Brain 2005b). The respondents were engineers, and had little trouble understanding the examples, given the documentation.

## 8. Results

### 8.1. COMPUTATIONAL RESULTS

Tables 2 to 5 show the averages of the computational results for the various measurements in the experiment. Higher values indicate a stronger match.

TABLE 2. Average of average normalised group.

| | **Low detail** | **High detail** |
|---|---|---|
| **EC** | 0.5543 | 0.4705 |
| **DC** | 0.6907 | 0.5580 |
| **BOTH** | 0.4390 | 0.3935 |

TABLE 3. Average highest normalized gmap weight per match.

| | **Low detail** | **High detail** |
|---|---|---|
| **EC** | 0.7629 | 0.6460 |
| **DC** | 0.6907 | 0.6086 |
| **BOTH** | 0.7081 | 0.6193 |

#### *8.1.1. DC and EC Comparison*

Our hypothesis concerning gmap weights was that the DC weights would be higher than EC weights. This is true for average gmap weight, but not true for highest gmap weight. The difference for low detail result, Table 2, is statistically significant ($p<0.05$), while the difference for high detail is not.

TABLE 4. Average standard deviations of normalized gmap weights per match.

| | Low detail | High detail |
|---|---|---|
| **EC** | 0.1796 | 0.1212 |
| **DC** | 0.0 | 0.0435 |
| **BOTH** | 0.2512 | 0.1275 |

TABLE 5. Normalized number of gmaps per match.

| | Low detail | High detail |
|---|---|---|
| **EC** | 0.9421 | 2.5664 |
| **DC** | 0.2952 | 1.2389 |
| **BOTH** | 0.4883 | 1.9624 |

This can be explained by the standard deviations in Table 4: it shows that the standard deviation for EC is higher than it is for DC. The standard deviations are statistically different ($p<0.05$). Although the EC representation might have a few gmaps with higher weights, it has other lower weighted gmaps that decrease the match's average gmap weight. Thus, in the experiments DC representations produced a few high weighted matches that have similar weights while, the EC representations produced matches that have a wider variety of weights. This resulted in lower average gmap weights and higher highest gmap weights for the EC representation.

Another one of our hypotheses was that EC would produce more matches than DC. The data, shown in Table 5, shows that EC produces at least twice as many gmaps as DC. This result is statistically significant ($p<0.05$).

*8.1.2. BOTH Dataset*

Our final hypothesis is that the matches from the BOTH dataset will have more matches of higher weights than the DC or EC datasets. This makes sense because the more information the representation has, the more it should be able to match.

Our results show the hypothesis is correct for absolute gmap weights, but not for the normalized weights. The normalized weights measure how much of the representation was matched. This result means that a large portion of the BOTH representation is left unused in each gmap.

We observed that gmap weights from the BOTH dataset have a lower highest gmap weight than the ones from the EC dataset and only a slightly higher highest gmap weight than the ones from DC dataset. The EC dataset has statistically different highest gmap weights and the DC dataset does not have statistically different highest gmap weights.

We also observed that the average gmap weight for the BOTH dataset was lower than it was for the DC and EC datasets. This effect is partly caused by the fact that BOTH has a higher standard deviation than DC.

However, this does not explain the difference the BOTH dataset has with the EC dataset, because they have about the same standard deviation. A statistical test did not reject the possibility that the standard deviations are similar.

One explanation for this is that when DC and EC information are together the DC information is preventing the matches that would have been generated if only the EC information was present. It could be that with the BOTH representation it is harder to make globally consistent gmaps, as there is so much data with which to be globally consistent. Because the normalization discounts for not having large matches, the match weights are lower.

Another observation about the BOTH dataset is that its highest gmap weight and number of gmap measures are in between DC and EC measures. It seems that adding EC information to the DC information improved the highest gmap weight and number of gmaps by only 24% to 55% of what would have been gained by using the EC information only. Investigating this further, we found that the number of gmap weights from the BOTH dataset is not statistically different from a dataset made by averaging the number of gmap weights from the DC and EC datasets. The average number of gmaps for the averaged DC/EC dataset is 1.8245, which is close to the value of 1.9624 for BOTH.

#### *8.1.3. Robustness to level of detail*

With a few exceptions, these observations are robust to changes in the detail of the representation. The data shows that the same trends occur in the low detail as in the high detail data. The observations that are different are caused by special properties of the low detail data. One difference is that the DC representation seems to be less effective in low detail devices than in high. The low detail DC representations produced one gmap at most for any matches. The high detail representation, however, did not have this problem. We conclude that the low detail representation is too small for our DC representation.

### 8.2. HUMAN RESULT ANALYSIS

We collected data from 10 respondents: a repertory grid and a questionnaire for each respondent. This section offers the results from this limited survey.

#### *8.2.1. Repertory Grid*

We use the "percent similar" measure generated by the repertory grids collected from the respondents, and compare that measure to the normalized highest gmap weight generated by SME. Each repertory grid was made between 6 devices, making 36 possible evaluations between devices.

Although percent similar can range between 0 and 1, it should not be directly compared to SME data since the repertory grid collection technique asks for clarification when similarity levels are above a certain percent. This makes the percentages artificially low. Therefore, we compute *match rankings* based on the percent similar measures generated by SME and the repertory grid.

We looked for correlations between the DC and EC data sets and each of the 10 individual respondents. We use the Spearman rank order test to detect correlation between the datasets and the Wilcoxon signed rank for testing that the medians of the differences between the datasets are different.

The tests show no significant correlations between the DC and EC datasets and the respondents' answers ($p>0.23$). The data also shows that the datasets and the respondents' answers are significantly different ($p<0.1$).

The repertory grid is also used to try to determine whether the respondents' reasons given in the repertory grid correlate with the SME datasets. We classified each of the respondent's constructs used in their repertory grid as DC, EC, or neither.

One of the respondents with the most EC constructs had the strongest correlation with the EC dataset. The respondent with the most DC constructs was slightly more correlated with the EC dataset. The respondent that was most correlated with the DC dataset had 4 EC constructs and 2 DC constructs. Our analysis also shows that sometimes, the classification of the respondents' constructs predicts which dataset they will be more correlated with. This occurred in the data from 5 of the 10 respondents.

*8.2.2. Questionnaire*

The Questionnaire consisted of 8 questions about novelty. Four of the questions were from DC matches and the other 4 were from EC matches. The questions spanned matches that SME gave high and low match weights to: i.e., high m-weight questions and low m-weight questions. Overall, the respondents marked 21 with high novelty, 30 with medium, and 29 with low.

First, we expected that EC matches would be more novel because EC can make a wider variety of matches. However, we discovered that the respondents considered the DC matches slightly more novel. Twelve of the 21 high novelty scores were for DC.

Second, we expected that the lower the SME match weight, the more novel the respondents would rate the match. Since a lower weight means that the match was not a very strong match, we expected that lower weighted matches would seem more original to the respondents.

The respondents' data shows this effect, Table 6. There were 5 high m-weight questions and 3 low m-weight questions. Nine of 21 high novelty ratings were given to the low m-weight questions for an average of 3 high novelty ratings per low m-weight question and 2.4 high novelty ratings per high m-weight question.

TABLE 6. Average number of novelty ratings per question class.

| | **High novelty** | **Medium novelty** | **Low novelty** |
|---|---|---|---|
| **High m-weight q's** | 2.4 | 4.4 | 3.2 |
| **Low m-weight q's** | 3 | 2.6 | 4.3 |

## 9. Discussion

The purpose of this research is to explore the differences between DC and EC representations of function. To do this we created a KR and represented a set of clock test examples. We performed a computational experiment with SME and performed an informal human experiment. From these we have discovered some properties of DC and EC representations that may be useful for computer-based design systems and the designers who use them.

First, our experiment shows how a designer might use a knowledge representation more effectively to generate novel matches. Our human experiment shows that the respondents determined low weighted matches to be more novel than high weighted matches. Our computational experiment shows that EC representations produce the most matches, some of which are low weighted. This suggests that to find novel matches a designer should prefer representations that are EC.

In addition to this, our computational experiments show that to make novel matches, the designer should not mix strictly DC representations and EC representations for several reasons. First, the experiments show that although the matches from the BOTH representation were as varied as EC, there were not as many. The experiments also show that adding extra DC information to EC representations causes them to perform worse than the EC representation alone.

Another way to produce novel matches is to use a DC representation alone. Our human experiment shows that the respondents rated matches that came from DC representations as being more slightly novel than matches from EC representations.

Unfortunately, our results are inconclusive about whether DC or EC representations are more useful for generating novel matches. On one hand, the low weighted matches that EC representations create can generate novel results. On the other hand, DC representations, which produce few low weighted matches, can also produce novel matches. Thus, more work needs to be done in order to determine which has a greater effect on producing novel matches.

Another result is that DC representations are useful for when the designer is looking for a few strong matches. By using a DC representation, the designer can expect to get fewer matches to sort though, and to find matches that are more relevant to their work. Chandrasekaran and Josephson (2000) say that it may be beneficial for designers to switch focuses from EC to DC at a certain point in the design process. This research suggests that this decision point may be when the designer wants the design system to produce fewer, more focused matches.

## References

Balazs, ME: 1999, *Design Simplification by Analogical Reasoning*. Ph.D. thesis, Worcester Polytechnic Institute, Computer Science Department, Worcester, MA, USA.

Besemer, SP and Treffinger, DJ: 1999, Analysis of creative products: Review and synthesis, *in* GJ Puccio and MC Murdock (eds) *Creativity Assessment: Readings and Resources*, Creative Education Foundation Press, Buffalo, NY. pp. 59-76.

Brain, M: 2005a, *How Digital Clocks Work*, http://home.howstuffworks.com/digital-clock.htm

Brain, M: 2005b, *How Pendulum Clocks Work*, http://home.howstuffworks.com/clock.htm

Boden, MA: 1994, What is creativity?, *in* MA Boden (ed), *Dimensions of Creativity*, MIT Press, Cambridge, MA, pp. 75-117.

Brown, DC: 1992, Design, *Encyclopedia of Artificial Intelligence*, John Wiley, 2nd ed, pp. 331-339.

Brown, DC and Blessing, L: 2005, The relationship between function and affordance, *Proc. ASME Design Theory and Methodology Conference*, DETC2005-85017.

Chandrasekaran, B and Josephson, JR: 2000, Function in device representation, *Engineering with Computers* **16**: 162-177.

Falkenhainer, B: 2005, *Structure Mapping Engine Implementation*, http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasoning/analogy/sme/0.html

Falkenhainer, B, Forbus, K and Gentner, D: 1989, The structure-mapping engine: Algorithm and examples, *Artificial Intelligenc*e **20**(41): 1-63.

Gentner, D, Holyoak, K and Kokinov, B (eds): 2001, *The Analogical Mind: Perspectives from Cognitive Science*, MIT Press, Cambridge, MA.

Goel, A: 1997, Design, analogy, and creativity, *IEEE Expert* **12**(3): 62-70.

Hart, A: 1986, *Knowledge Acquisition for Expert Systems*, McGraw-Hill, New York.

Kitamura, Y, Kashiwase, M, Fuse, M and Mizoguchi R: 2004, Deployment of an ontological framework of functional design knowledge, *Journal of Advanced Engineering Informatics* **18**(2): 115-127.

Pahl, G and Beitz, W: 2003, *Engineering Design: A Systematic Approach*, Springer, 2nd ed.

Prabhakar, S and Goel, A: 1996, Functional modeling of interactions between devices and their external environments for adaptive design, *Proc. Modeling and Reasoning about Function workshop*, AAAI-96 Conference, Portland, Oregon, pp. 95-106.

Rosenman, MA and Gero, JS: 1998, Purpose and function in design: From the socio-cultural to the techno-physical, *Design Studies* **19**(2): 161-186.

Shaw, LG and Gaines, BR: 2005, Rep IV 1.10. http://repgrid.com, Cobble Hill, BC, Canada

Stone, RB and Chakrabarti, A (eds): 2005, Engineering applications of representations of function, *AIEDAM* **19**(2&3): 359-370.

Stone, RB and Wood, KL: 1999, Development of a functional basis for design, *Proc. ASME Design Theory and Methodology Conference*, DETC99/DTM-8765.

Umeda, Y and Tomiyama, T: 1997, Functional reasoning in design, *IEEE Expert* **12**(2): 42-48.