ABSTRACT

Industrial accidents continue to happen despite rapid technological advancement and they are often caused by triggers similar to those of past accidents. If we turn our eyes to the world, especially to the emerging industrial players, we hear news about accidents caused by phenomena that have already caused similar accidents elsewhere.

Industries, as they emerge and grow over hundreds of years, learn their lessons throughout their histories and build rules, regulations, and common knowledge to avoid accidents. Each industry is probably well aware of accidents that took place in its own country, especially when the accident led to enforcement of a new law. Nevertheless, we hardly have any knowledge of accidents in foreign countries unless they were of huge sizes.

Japan had a national project of building a database of knowledge and lessons learned from past accidents. Failure Knowledge Database (FKDB) went on the Web in 2005. As of today it still attracts a large number of readers with its over 1,600 failure cases. Our research is targeted at making use of this FKDB by abstracting the knowledge, especially what triggered the accidents, and comparing the knowledge with functional and structural elements used in new designs.

Design Record Graph (DRG) is a graphical representation of the designer’s intention starting from the left with the product functional requirement which iteratively divides into sub-functions to reach a set of functional elements (FE). Each FE maps to a structural element (SE). Then the SEs iteratively combine to form assemblies and finally the product at the right end. A failure starts from one of the FE-SE pairs and propagates the DRG in both left and right directions to reach the two ends. The propagation leaves a trace of how the point of failure led to disabling the product.

For each failure case in FKDB, we identified the origin of failure, the FE-SE pair that started the accident. An FE is abstracted by a verb phrase and a set of noun phrases, and similarly an SE with some noun phrases. By limiting the phrases to use, similar concepts are described by the same abstracted phrases.

A new design has a number of FE-SE pairs and their propagations in the DRG to reach the two ends. The designer can then compare all propagations in the design, without the knowledge if any of them are dangerous, with those in FKDB that are known to have led to accidents.

We developed quantitative operators to evaluate the similarity between two traces. Our results offer a way of warning the designer about possible flaws in a new design similar with causes of past accidents that the designer has no idea about. Our method of preventing design failure can apply to other fields for novice planners in avoiding failure while still in the planning stage. We can further develop the use of knowledge into overseas countries by mapping the limited number of verb and noun phrases into foreign language.

1. INTRODUCTION

Advancements in technology have now made it possible for us to produce machines with great complexity. The machines we design and manufacture grew in size, weight and complexity, thus, assuring safety with them has turned into an essential task in designing, producing, maintaining, and using them [1]. With new and old machines surrounding our lives, each industry has gone through its learning curve of what can cause accidents. Typical machines today have their roots in the industrial revolution that started in the 18th century. Thus they
have histories of some hundreds of years. Recent advancement of information processing with the network and computers have forced our machines into another complexity.

The efforts of emerging countries have changed the world map of what industries are located where. The learning curves in these regions are much steeper compared to when the industries were first formed. The newcomers have samples to follow.

This fact, however, tends to push industries in emerging countries to quickly turn profitable, to follow the design forms without knowing the design philosophy of "why the part is shaped that way."

In March of 2004, a boy was caught in a revolving door by his head and lost his life. The revolving door, in Roppongi Tokyo, was a large one with a 4.8m diameter. The original design, made in Netherlands, had a revolving part that weighed only a ton. The revolving part of the accident door weighed 2.7tons with motors and breaks mounted on it with stainless steel lining and tempered glass for its looks. There was a reason for the original design to keep the revolving part light, however, that philosophy was lost in the new design that looked better.

In August of 2015, a series of explosions took place at the port of Tianjin, China. The location was where industries stored materials for chemical processing. Firefighters rushed to the location and started pouring water on the fires. The warehouses in the area had chemicals like sodium cyanide (produces toxic hydrogen cyanide gas when wetted with water or acid) and calcium carbide (releases highly volatile acetylene gas when it meets water) among other highly explosive nitric compounds.

Pouring water over the fires worsened the accidents, however, if left untouched, the nitric compounds would have exploded anyway. What could have been avoided is the death of 104 firefighters among the total casualty count of 173 [2].

In July of 1964, a fire broke out in the Port of Tokyo, about 7km south from Tokyo Station. Firefighters rushed to the scene. Unaware that highly explosive methyl ethyl ketone peroxide (MEKPO) was in the adjacent warehouse, 19 firefighters were killed when it exploded [3]. This accident led to the revision of the Fire Service Act to enforce mandatory inspection approval for storage of hazardous material.

The tragedy in Tianjin could have been avoided if the firefighters had the knowledge that pouring water over a fire in a chemical storage compound was dangerous. Although the lesson we learn from this case is different from a design case on a table, it is rooted in the same point that better international knowledge communication could have prevented the disaster. The functional requirement from a designer's view was to “stop fire from spreading” with a solution of “pour water.” Firefighters in Japan today when faced with fire in a chemical plant will rush to the scene but will keep the fire engines some distances away and will not pour water over the flames until they confirm what are in the area with responsible plant personnel.

This paper proposes a step forward in better transfer of knowledge about past accident cases to new players in the industry and novice designers alike. If the designers take the time to thoroughly study accidents in the field, many of the design flaws and tragic accidents will be avoided. The average designer, however, is constantly chased by work and hardly has the time to study failure cases. Especially those from emerging countries have difficulty with language barriers.

Our method takes accident cases from a large knowledge base and identifies what triggered the accidents. An accident typically starts from a structural element (SE) failing to meet its functional element (FE). We then abstract the intention when the designer worked the troubled configuration into the design. Information of a failure case is abstracted in the form of “what the designer intended to do,” instead of “what went wrong.”

When a designer completes a new design, the set of FE and SE pairs can be compared with those with records of unintended failures with past designs. The designer will then be alerted which part of the just finished new design can have possible problems that a former designer planned similarly but without an idea that the concept could fail.

We learn much from accidents and troubles that take place in our own country. We, however, hardly learn from accidents overseas unless their scales are catastrophic to make international news. New players in the industry follow quick learning curves but they do not need to repeat the same mistake that took place elsewhere. The designer will spend little efforts to describe new designs with our graph representation and then looking up the knowledge base for similar failure cases is automatic.
2. RELATED WORK

Tools for detecting parts of design that can lead to failures exist, e.g., fault tree analysis (FTA), failure mode and effects analysis (FMEA) [4], or System Theoretic Process Analysis (STPA) [5]. I-TRIZ stimulates the designer to find potential failures in designs by posing the question “how can the system accomplish the failure?” [6]. Nevertheless, problematic designs still manage to escape the design efforts if the design team have limits in its experience and imagination.

Stone and Wood developed a functional basis, a standard set of functions and flows, for describing the mechanical design space [7]. The paper extensively lists other work in defining standard functional terms. The readers are encouraged to read it to learn the background in developing a functional basis. Tumer and Stone developed the function-failure theory to alert the designer about potential failure of functions in a new design [8]. The method identifies potential failure information for a set of functions in a new design using known component-failure and function-component relations. For the purpose, standardized vocabulary sets for functions and failure modes are used. The knowledge base with this method grows as more cases are fed to the component-failure matrix [9]. The concept has led to a number of research results, one of which even generates recommendations about new designs [10].

The National Institute of Standards and Technology (NIST) separately worked on developing a formal representation to openly share across many design activities [11]. The two efforts met to produce a reconciled functional basis [12]. The motivation for this work was to build an unbiased open formal representation in the modern design environment that is heterogeneous and spread over the world [11].

For evaluating product similarity, McAdams, Stone, and Wood reported a method relating sub-functions of a product to customer needs [13].

3. OVERVIEW AND BACKGROUND OF OUR WORK

The work we report in this paper also aims at alerting the designer about potential problems that were overlooked during conceptual design. We also use a knowledge base of past failure cases and sets of standardized vocabulary. One of the vocabulary sets is of functional verbs and the other is of nouns or objects in the verb-object pairs and structures the designer adopted to meet functional requirements.

Points that distinguish our work from prior work are:
- We store knowledge about a failure only at the point of failure, i.e., what structural element (SE) failed to meet which functional element (FE) and two higher level functional requirements. The difference with function-failure theory built on functional basis [8] is that function-failure theory defines failure modes of components in a failure mode vector for each component, whereas, we have a single failure mode for a specific component in each failure case. In other words, our failure mode vector is scattered over the FKDB that continues to grow as failure case reports are added.
- Making contribution to the knowledge base is easy because one has to provide the FE-SE pair and two upstream functions at the point of failure described with verbs and nouns from the standard lists. We do not have to identify all failure modes of a specific component, like in [8, 9, 10].
- For a new design, when complete, the designer selects verbs and nouns for all FE-SE pairs in the design. Then the system automatically compares similarity of all these FE-SE pairs with those that triggered failure cases stored in the knowledge base.
- Similarity evaluation of an FE-SE pair in a new design with one from the failure knowledge base is made by counting how far, in the verb hierarchy, apart the verb in the new design is from that in the known failure case. A similar evaluation is made for nouns in the FE-SE pairs and added to the result.
- Instead of trying to establish formalized sets of terminology, we leave the verb and noun hierarchies open to alterations by the project members. There is, however, supervision by committee members of the group. The group consists of engineers and non-engineers across Japan working for different industries and companies.
- If one keeps a list of own designs, when a new failure case is added to the knowledge base, it can trigger a new comparison and if applicable, can trigger an alert. This feature is important in identifying troubles with old designs.

The motivation for our method is that at the time of completing a design, the designer has no idea of what failure modes may have been worked into the design.
Japan Science and Technology Agency (JST) of the Ministry of Education, Culture, Sports, Science and Technology (MEXT) started collecting accident information in 2002. The archive work made the information publicly available on the Internet since March of 2005. The knowledge base is now managed by Hatamura Institute for the Advancement of Technology and is named Failure Knowledge Database (FKDB) [14]. The knowledge base has descriptions of over 1,600 industrial accident cases.

In the past, we characterized each failure case by describing its scenario; the three steps of cause, action, and results with a sequence of phrases from a given set [15]. Defining scenarios for failure cases was effective in organizing the cases and identifying those with similar causes with others. The method, however, still relied on the designer’s efforts to read through the cases to gain knowledge of past failures and recognize similarities with new designs.

The next iteration in our efforts sought a way of alerting the designer about possible causes of failure in new designs without the designer having to actively look through repositories of past failure information. We explained the principles of our method [1], however, had not built a tool for applying it. This paper reports the tool we developed and its outcome.

4. DESIGN RECORD GRAPH

We earlier called this concept, function-structure diagram (F-S Diagram) [1] following the Stanford naming of function-structure diagram [16,17]. Hatamura and Nakao separately developed the same concept [18,19]. To avoid confusing our diagram with the functional model [7], that has input and output identified for each function, we call ours Design Record Graph (DRG) because it is intended more for describing the designer’s development of design starting from the product functional requirement.

DRG is a directed graph with functional nodes in the left and structural nodes in the right. The two sides respectively show the hierarchy of functions and structures.

The left-most functional node is the product functional requirement, also called the maximum functional requirement. The right end is the product. This graph representation, often used for mechanical design, is simple and easy to use, thus, it is also frequently used for other applications like service engineering, product planning and software development.

The product functional requirement divides into sub-functions and continues so until the functions are divided into a set of FEs. An FE maps across the border to the structure space to one or more SEs. The SEs gather to form components and higher level assemblies until the right end product is reached.

![Design Record Graph](image)

Fig. 1 shows a typical DRG. Note that there are some irregular correspondences; FE to SE mapping is not necessarily one to one, and a child node may have more than one parent.

When a design fails, the starting point is an SE failing to meet its FE. This is the failure mode that causes an accident. The designer does not expect such a failure to take place at the time of design construction.

For past accidents, the DRG is not readily available, however, failure analysis allows us to reconstruct the designer’s intention around the point of failure. The failure then propagates DRG to the left through the function space and to the right through the structure space, and the product fails to meet its product functional requirement. We call this path the failure path of the accident [1]. Fig. 2 shows where the failure mode is and its propagation through DRG to define the failure path.
5. EVALUATING SIMILARITY BETWEEN TWO PATHS

Track of failure and track of design

FKDB stores failure case information for over 1,600 accident cases. The knowledge of each case is associated with the failure path of the design (Fig. 2). Thus, we have over 1,600 failure path information abstracted to “track of failure,” described after Fig. 3, for comparison with a new design. Note that the failure case analyzer does not need to define the whole DRG of the failed design.

For a new design, the designer has a complete DRG. Fig. 3 shows an example with 6 functional and 6 structural nodes. This design has 3 FE-SE pairs, 4-7, 5-8, and 6-9. These pairs give the 4 tracks of design that could possibly fail as Fig. 3 shows.

We define a track of design and of failure as follows:

A link that crosses the functional space to the structural space is the origin, an FE-SE pair, of a track. There is one failure track for a failure case, and multiple design tracks for a new design. A track consists of 3 functional nodes, the FE and 2 nodes upstream of the FE (to the left in Fig. 3), and the structural node SE.

The design in Fig. 3 thus has 4 tracks as the lower part of the figure shows. These 4 tracks define the potential failure modes of the new design.

The reason for taking two upstream functional nodes in addition to the FE is to cope with subjectivity of the designer or analyzer in defining the nodes. Some may analyze a failure case to the fine details or define a complete DRG to the nuts and bolts for a new design. Others may stop their analysis at a more abstract level for elements of function and structure. If we look only at the FE and SE, we could easily miss a matching case.

On the other hand, assemblies upstream of the SE (to the right in the structural space) may very well be different even if concepts at the element level are identical, thus, we only include the SE of a path in the track.

The designer, when the design is complete, has a number of tracks of design. The system automatically compares each one of them with the tracks of failure stored in the FKDB. Next we describe the phrase lists we prepared and then similarity evaluation of two tracks.

Verb and noun phrase lists

Each track has 3 functional nodes and a structural node. For abstracting a track, we prepared lists of verb phrases (VP) and noun phrases (NP). The system displays these lists to the designer or failure case analyzer. These standard lists are fixed at any time, however, they remain flexible to change in their hierarchical structure and components.
Members of our project are free to propose changing hierarchical position of phrases (VP or NP), to add new ones or merge multiple ones. Any such proposals are evaluated by the committee for acceptance or denial. Additions of new phrases do not affect past design and failure analysis, however, changes in the hierarchy will cause changes in past analysis results. Hierarchical changes made so far, however, have made small changes to past results.

This way of maintaining our phrase hierarchies was inspired by how Wikipedia maintains its articles. The lists are dynamically defined with group efforts instead of someone at some point, declaring what they should be.

Annex A shows partial lists of VP and NP with phrases used in this paper. The design or analyzer picks out one VP and one or more NPs to describe a functional node. For a structural node, i.e., the SE, the designer picks out one or more NPs.

In most cases, the number of noun phrases needed to describe a functional or structural node is one, however, two or more may be appropriate depending on the node. Fig. 4 shows a track and the phrases that abstractly describe nodes of the track.

Note that the description for each node, e.g., “Adhere anchor bold to hole” in Fig. 5, is not confined to using verbs and nouns from the limited phrase hierarchy. The designers and failure case analyzers are free to describe their nodes with their words. The verb and noun phrases (VP and NP), however, have to be selected from the hierarchical lists of them.

```
Adhere anchor bolt to hole
```

VP: glue
NP1: anchor bolt
NP2: oversized hole

**Fig. 5 Functional node, its description, VP and NPs**

**Affinity of phrases, similarity of nodes, and of tracks**

We now describe the operators we developed for evaluating similarities between two tracks.

First we define the distance operator $Dist(a, b)$ to mean the distance between two phrases $a$ and $b$ in the phrase hierarchy. When we connect the two phrases $a$ and $b$ by the shortest path, we can count the number of phrases in this path including the two ends $a$ and $b$. This count gives the quantity $Dist(a, b)$. Thus, when the two phrases are identical, the count $Dist(a, b)$ is 1, and it is 2 when the two are immediately adjacent. The affinity operator $Aff(a, b)$ is simply the inverse of $Dist(a, b)$. Fig. 6 shows some examples of how $Aff(a, b)$ evaluates.

\[
Dist(a, b) = (\text{number of phrases in the shortest link between phrases } a \text{ and } b \text{ in the phrase hierarchy}) \quad (1)
\]

\[
Aff(a, b) = Dist^{-1}(a, b) \quad (2)
\]
Now we define a track of design TD with nodes D₁, D₂, D₃, D₄ and a track of failure TF with nodes, F₁, F₂, F₃, F₄. The suffix numbers 1 through 4 follow the system in Fig. 4. Then we define the similarity ($\sigma$) between two tracks TD and TF as follows:

$$\sigma(\text{TD}, \text{TF}) = \sum_{i,j=2..4} \text{Sim}(D_i, F_j)$$

where similarity of nodes, $\text{Sim}(D, F)$ is defined:

$$\text{Sim}(D,F) = \text{Aff}(\text{VP}_D, \text{VP}_F) + \frac{1}{n m} \sum_{i=1..n, j=1..m} \text{Aff}(\text{NP}_D^i, \text{NP}_F^j)$$

where $n$ and $m$ are the number of noun phrases NP for nodes D and F, respectively. The first term of Eq. (4) is 0 when evaluating the similarity of SE nodes because they do not have verb phrases.

The similarity of two tracks shall not be larger when nodes have multiple noun phrases. Thus, in evaluating similarity of two nodes, all noun phrases from one node are evaluated for their affinities with all noun phrases from the other, and we divide the resulting quantity contributing to similarity by the product of the number of noun phrases from the two nodes.

Lastly, for the 4 nodes of a track in Fig. 4, we assigned weights, 4 for Node 1 and Node 2, 2 for Node 3 and 1 for Node 4. The reason is because similarity of nodes closer to the border between the functional and structural spaces shall contribute more to similarity of tracks. These low level nodes describe the actual method (SE) of realizing the precise need (FE) of a design. The choice of this ratio 4, 2, 1 is simply because we chose to use 4 nodes in three levels next to the border of functions and structures, and cutting the node contribution to similarity in half as the nodes moves away from the border seemed reasonable. We thus have a modified equation for Eq. (4), with consideration to node position dependent weight.

$$w\text{Sim}(D,F) = w_D w_F \text{Aff}(\text{VP}_D, \text{VP}_F) + \frac{w_D w_F}{n m} \sum_{i=1..n, j=1..m} \text{Aff}(\text{NP}_D^i, \text{NP}_F^j)$$

where $w_D$ and $w_F$ are 4 for nodes in position 1 or 2, 2 for nodes in position 3, and 1 for position 4.

The maximum value of phrase affinity is 1 (Fig. 6) when the two phrases are identical. The phrases used for adjacent nodes are most likely to be different except for some rare cases. So the least distances among the three nodes in the function space are 3, 2, and 1 as Table 1 shows.

<table>
<thead>
<tr>
<th>Dist</th>
<th>Phrase 4</th>
<th>Phrase 3</th>
<th>Phrase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phrase 4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Phrase 3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Phrase 2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Then the largest affinity values for the three phrases (verb or noun) are as Table 2 shows. Note that multiple phrases for the noun section do not affect the similarity evaluation because their contributions are divided by their counts at the end.
When we introduce the weights, 4, 2, and 1 into the quantification, we obtain Table 3 for the largest affinity values (wAff) for the three nodes in the function space.

### Table 3 Largest weighted Aff of phrases in function space

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Phrase 4</th>
<th>Phrase 3</th>
<th>Phrase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aff</td>
<td>1</td>
<td>1/2</td>
<td>1/3</td>
</tr>
</tbody>
</table>

The 9 wAff values in Table 3 sum to 33.67. This is the largest contribution from the three verb phrases in the function space to the first term in Eq. (5). The set of three noun phrases make the same maximum contribution because their multiplicity cancel with the division as shown in Eq. (5).

The first term of Eq. (3), i.e., the similarity of the two noun phrases has the largest possible value of 16 because the two phrases for D₁ and F₁ both have weights 4, and the affinity of two nodes is maximized at 1. Adding these results, we find that the maximum value for similarity between two tracks is:

$$\sigma_{max} = 16 + 2 \times 33.67 = 83.34$$  \hspace{1cm} (6)

### 6. SAMPLE CASE RUN

For a sample failure case to reference against a new design, we picked the “2006 Sasago tunnel, ventilation duct ceiling slab collapse” case [1]. The DRG for the case is shown in reference [1]. We defined the track of failure and picked out the verbs and nouns from the lists in Annex A as Fig. 7 shows.

![Fig. 7 Track of failure and verb and noun phrases for the Sasago Tunnel concrete slab collapse case](image)

The following is a sample new design case:

When we walk the villages in Switzerland, many of the tourist attraction buildings and hotels have their windows decorated with flowers. Buildings in the city of Tokyo have concrete and mortar walls decorated with neon without the flavor of nature and its wonders. We would like to decorate the windows of buildings in Tokyo like those in Switzerland. Fig. 8 shows our new design for planters to install under windows on concrete walls. Fig. 9 shows the DRG for this new design.
The DRG in Fig. 9 shows 6 potential failure modes that cross the border between functional and structural spaces. From the corresponding 6 tracks of design, we will just show the results from the most suspicious track, however, note that the designer, without knowledge in engineering design, will run all 6 cases against the failure knowledge base.

Fig. 10 shows the third track of design from the top of Fig. 9 and the verb and noun phrases picked out from Annex A.

Now we have two tracks with their abstracted phrases for similarity evaluation. Table 4 shows the computation of distances $\text{Dist}$ (from Annex A), affinities $\text{Aff}$ (inverse of $\text{Dist}$), and weighted affinities $w\text{Aff}$ ($\text{Aff}$ multiplied by $w_D$ and $w_F$). The total in the third table 12.38 is the verb phrase contribution to similarity from the functional nodes.
Table 4  Verb phrase contribution of comparing planter design and tunnel failure

<table>
<thead>
<tr>
<th></th>
<th>Dist</th>
<th>Suspend</th>
<th>Fix bolt</th>
<th>Glue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Fix bolt</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Grasp</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Next, we evaluate the contributions from the noun phrases in the tracks. Node 1 from the two tracks are evaluated only against each other. A node in the functional space is evaluated against the three in the functional space in the other track. Table 5 shows the Dist values found from Annex A.

Table 5  Noun phrase distances in the two tracks

<table>
<thead>
<tr>
<th></th>
<th>Dist</th>
<th>Threaded rod</th>
<th>Ceiling Plate</th>
<th>Ceiling Anchor bolt</th>
<th>Oversized hole</th>
<th>Epoxy glue</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Dist</th>
<th>Threaded rod</th>
<th>Ceiling Plate</th>
<th>Ceiling Anchor bolt</th>
<th>Oversized hole</th>
<th>Epoxy glue</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 6 shows Aff, the inverse values of Dist. Finally, the affinity values are multiplied by their weights to produce contribution values wAff shown in Table 7.

Table 6  Noun phrase affinities in the two tracks

<table>
<thead>
<tr>
<th></th>
<th>Aff</th>
<th>Threaded rod</th>
<th>Ceiling Plate</th>
<th>Ceiling Anchor bolt</th>
<th>Oversized hole</th>
<th>Epoxy glue</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.14</td>
<td>0.33</td>
<td>0.17</td>
<td>0.33</td>
<td>0.14</td>
<td>0.14</td>
</tr>
<tr>
<td>3</td>
<td>0.14</td>
<td>0.33</td>
<td>0.17</td>
<td>0.33</td>
<td>0.14</td>
<td>0.14</td>
</tr>
<tr>
<td>2</td>
<td>0.10</td>
<td>0.13</td>
<td>0.11</td>
<td>0.13</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>0.33</td>
<td>0.14</td>
<td>0.17</td>
<td>0.14</td>
<td>0.33</td>
<td>0.20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Aff</th>
<th>Threaded rod</th>
<th>Ceiling Plate</th>
<th>Ceiling Anchor bolt</th>
<th>Oversized hole</th>
<th>Epoxy glue</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.14</td>
<td>0.17</td>
<td>0.14</td>
<td>0.20</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.14</td>
<td>0.17</td>
<td>0.14</td>
<td>0.20</td>
<td>1.00</td>
<td>0.14</td>
</tr>
<tr>
<td>2</td>
<td>0.14</td>
<td>0.17</td>
<td>0.14</td>
<td>0.20</td>
<td>1.00</td>
<td>0.14</td>
</tr>
<tr>
<td>1</td>
<td>0.14</td>
<td>0.17</td>
<td>0.14</td>
<td>0.20</td>
<td>1.00</td>
<td>0.14</td>
</tr>
</tbody>
</table>

10
Table 7 Noun phrase weighted similarities in the two tracks

<table>
<thead>
<tr>
<th>wAff</th>
<th>Threaded rod</th>
<th>Ceiling</th>
<th>Pole</th>
<th>Ceiling</th>
<th>Anchor bolt</th>
<th>Oversized hole</th>
<th>Epoxy glue</th>
</tr>
</thead>
<tbody>
<tr>
<td>WF</td>
<td>12.38</td>
<td>14.23</td>
<td>26.61</td>
<td>83.34</td>
<td>10.75</td>
<td>13.90</td>
<td>4.00</td>
</tr>
</tbody>
</table>

When we sum the similarity contributions, we obtain 12.38 + 14.23 = 26.61. This is 32% of the full score of 83.34 computed with Eq. (6). A careful designer may be worried by this number and concerned with the anchor bolt getting pulled out. Fig. 11 shows a better design that significantly lessens the pull-out force on the anchor with the lever effect of the long arms.

![Improved concrete wall planter design](image)

**Fig. 11 Improved concrete wall planter design**

Next we show a case similar to the Sasago tunnel accident.

Fig. 12 shows a design employed more recently with higher performance fans for ventilating exhaust gas from tunnels. Fig. 13 is the DRG of the ventilation system and Fig. 14, the track of design that is likely to result in high similarity with the track of failure in Fig. 7. Note that the designer, without prior knowledge, will run all 6 tracks of the design against all known failure cases.

![Improved tunnel ventilation design](image)

**Fig. 12 Improved tunnel ventilation design**
Comparing the two tracks from Fig. 7 and Fig. 14, we gain the VP contribution in Table 8 and NP contribution in Table 9. Note that we intentionally introduced some differences with the selection of phrases and detail level.

The total similarity evaluation between the two tracks is now $23.58 + 30.83 = 54.41$. This is 65% of the full score 83.34. The designer is definitely alerted to look into the epoxy glue design and compare it with the failure case of Sasago tunnel.
When we carefully review the calculation, the exact match of the noun phrase “Epoxy glue” and verb phrase “glue” largely contributed to the high score. Limiting the phrases to use with our method will lead the novice designer to review the design against the Sasago tunnel case well documented in FKDB.

### Table 9 NP contribution to wAff calculation

<table>
<thead>
<tr>
<th>Design track node #</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dist (number of phrases in the shortest link from Annex A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threaded rod</td>
<td>Ceiling</td>
<td>Plate</td>
<td>Ceiling</td>
<td>Anchor bolt</td>
</tr>
<tr>
<td>4</td>
<td>Gas</td>
<td>8</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Fan</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Heavy object</td>
<td>11</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>Ceiling</td>
<td>7</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Anchor bolt</td>
<td>3</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>Oversized hole</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>Epoxy glue</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design track node #</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aff (Inverse of Dist)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threaded rod</td>
<td>Ceiling</td>
<td>Plate</td>
<td>Ceiling</td>
<td>Anchor bolt</td>
</tr>
<tr>
<td>4</td>
<td>Gas</td>
<td>0.12</td>
<td>0.17</td>
<td>0.14</td>
</tr>
<tr>
<td>3</td>
<td>Fan</td>
<td>0.14</td>
<td>0.14</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>Heavy object</td>
<td>0.09</td>
<td>0.13</td>
<td>0.11</td>
</tr>
<tr>
<td>1</td>
<td>Ceiling</td>
<td>0.14</td>
<td>1.00</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>Anchor bolt</td>
<td>0.33</td>
<td>0.14</td>
<td>0.17</td>
</tr>
<tr>
<td>1</td>
<td>Oversized hole</td>
<td>0.20</td>
<td>0.14</td>
<td>0.17</td>
</tr>
<tr>
<td>1</td>
<td>Epoxy glue</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 7. DISCUSSIONS

We developed a method for describing past failure accident cases abstractly with key phrases. The same abstraction method can apply to every possible point of failure of a new design. We then developed an operator to quantitatively evaluate similarity of possible points of failure with a new design to failure modes of past design.

When compared with the function-failure theory [8] that holds failure knowledge in a failure mode vector for each component, our method stores one element of the failure mode vector in a failure case track. We developed our method from the assumption that “the failure knowledge base developer can overlook failure modes of a component.” Allowing project members to contribute failure cases to the knowledge base and
propose changes to the verb and noun phase hierarchies keeps our knowledge base growing. Our method is a dynamic knowledge base system on the Internet.

The algorithm is straightforward except the part where the designer defines a DRG and associates VP and NP to the nodes near the functional and structural space boundary. In fact, even the same designer often draws slightly different DRGs on different days. The difference is small and the use of phrases from lists of limited phrases has the effect of keeping subjectivity caused variation small.

If a designer keeps past design records and their tracks of design, when a new piece of knowledge is added to FKDB, it can trigger an evaluation of the new information against the design records to alert dangers with old designs.

Our distance operator $\text{Dist}$ is a simple count of how far two phrases are located in the list of phrases, and affinity operator $\text{Aff}$ its inverse. This means, when two phrases are identical, $\text{Aff}$ is 1, and for parent-child relations 0.5. Siblings have affinity values of 1/3. If the noun phrase NP1 was a sibling instead of being identical for the fan case in Fig. 12, the total similarity score drops by about 10. Still high enough to alert the designer, however, cutting the affinity in half when two phrases are not identical but are adjacent may be penalizing the similarity by too much. Further experience with our method will lead to a more appropriate ratio to apply than one half. In other words, our weighing scheme of 4, 2, and 1 depending on the node position in the DRG needs some further experience to justify its appropriateness.

8. CONCLUSION

We developed a method for abstractly describing which designer’s intension failed in a failure case. We prepared these abstract descriptions, which are collections of verb and noun phrases with hierarchy and weights, for failure cases in a large knowledge base FKDB. When a designer constructs a DRG for a new design, all possible failure modes are the mappings from an FE to an SE. The abstraction method we developed can express all these possible failure modes also as a collection of hierarchical verb and noun phrases with weights. The abstraction enforces the use of phrases from sets of standard verb and noun phrases in phrase hierarchies. The phrase sets are continually modified as needs arise, however, are placed under our supervision for control.

We developed operators to quantitatively evaluate similarities between a failure case and a possible failure mode of a new design. Then a designer can evaluate all possible failure modes of a new design with those listed in FKDB.

This method allows the designer to identify possible problems with new designs even if the designer was unaware that such a failure case existed.

By keeping tracks of old design records, when a new knowledge is added to the knowledge base, the designer can evaluate the records against the new failure knowledge.

The failure mode description is currently available in English and Japanese. Porting the system to another language simply takes translating the verb and noun phrase lists. There is no need to translate the failure case descriptions. The verbs and nouns are hierarchical lists of phrases in their languages, however, in essence they are lists of verbal and noun concepts. It is our hope to reduce the number of accidents in the future in regions with less experience in industries, as well as in new designs that are challenging but maybe made by designers without enough experience.

Industries take decades to grow mature and all experiences during the growth are valuable lessons to avoid accidents in pursuing the industry. Introducing an industry in its mature form to a new region takes away the chances of learning the design philosophy and background of why a design is made that way.

Our development alerts designers when their new ideas have been proven dangerous with similarities with accidents taken place elsewhere. It helps the designer without experience or knowledge about past failure cases.
REFERENCES


ANNEX A

PHRASE HIERARCHY LISTS

VERB PHRASE LIST
Verb phrase (partial listing)
+ Provide
+ Prevent
+ Determine
  + Determine position
  + Determine relative positions between solids
    + Fix relative position
    + Adhere
    + Glue
    + Weld
    + Mechanically bind
      + Fix with bolt
      + Rivet
      + Support weight
      + Put on top
      + Suspend
      + Fix on vertical wall
      + Hold
      + Grasp
    + Form fluid path
      + Form strict path
      + Form open path
    + Force flow
    + Exhaust

NOUN PHRASE LIST
Noun phrase (partial listing)
+ Physical substance
  + Fluid
    + Gas
  + Solid object
    + Machine element
      + Connecting element
        + Threaded element
          + Bolt
          + Screw
          + Threaded rod
          + Anchor bolt
          + Hole element
            + Oversized hole
            + Anchor
          + Electrical machine
            + High power electrical machine
            + Fan
        + Standard element
          + Plate
      + Structural element
        + Pathway
          + Flow channel
          + Wall
            + Shielding wall
            + Ceiling
            + Window
            + Decoration
        + Structural element
      + Standard element
        + Plate
    + Material
      + Metal
      + Iron
      + Plastic
      + Concrete
    + Process
      + Mechanical connection
        + Fix with bolt
      + Adhesive connection
        + Glue
        + Epoxy glue
    + Adjective noun
      + Weight
        + Heavy object
        + Dead weight
    + Verbal noun
      + Motion
      + Fall
    + Conceptual noun
      + State description