2018-01-1200 Published 03 Apr 2018

Big Data Analytics for Improving Fidelity of Engineering Design Decisions

Baldur Steingrimsson Imagars LLC

Sung Yi and Robert Jones Portland State University

Mikalai Kisialiou Imagars LLC

Kwan Yi Eastern Kentucky University

Zachary Rose Portland State University

Citation: Steingrimsson, B., Yi, S., Jones, R., Kisialiou, M. et al., "Big Data Analytics for Improving Fidelity of Engineering Design Decisions," SAE Technical Paper 2018-01-1200, 2018, doi:10.4271/2018-01-1200.

Abstract

This paper presents a high-level framework (vision) for utilizing big data analytics to harvest repositories of known good designs for the purpose of aiding mechanical product designs. The paper outlines a novel approach for applying artificial intelligence (AI) to the training of a mechanical design system model, assimilates the definition of meta-data for design containers (binders) to that of labels for books in a library, and represents customers, requirements, components and assemblies in the form of database objects with hierarchical structure. Design information can be harvested, for the purpose of improving design decision fidelity for new designs, by providing such database representation of the design content. Further, a retrieval model, that operates on the archived design containers, and yields results that are likely to satisfy user queries, is presented. This model, which is based on latent semantic analysis (LSA), predicts the degree of relevance between accessible design information and a query, and presents the most relevant previous design information to the user. A simple example, one involving idea generation for conceptual design, is presented, in order to provide insight into the significant utility that may be derived from the proposed AI design framework.

Keywords

Big Data, Product Design, Semantic Analysis, Decision Making, Design Software

Introduction: AI Applied to Product Design

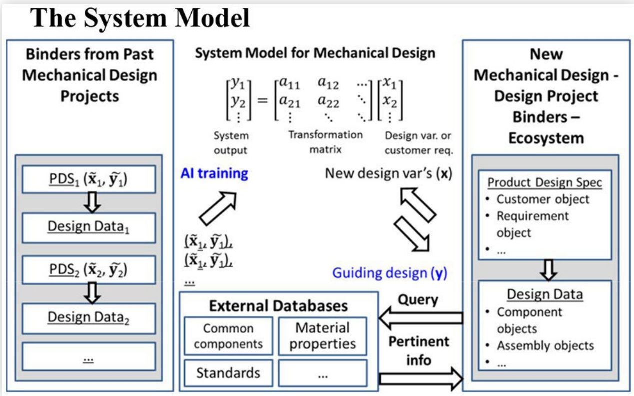
This paper presents a high-level framework for applying artificial intelligence for the purpose of aiding with mechanical product design. There presently is significant interest in big data analytics, especially within the automotive industry. Large amounts of data are collected from fleets of vehicles. The data is being uploaded to cloud systems, where it is analyzed using big data and machine learning algorithms. Then, information of interest can be communicated to drivers for system feedback. In addition, some streaming data can be made available to an automotive vendor for efficiency and maintenance monitoring, or used internally by an original equipment manufacturer (OEM) for post-mortem failure analysis.

With utilization of big data within the automotive industry on the rise, applications to the process of product design have still been limited. In [1], AI was used to improve the way that agents (people or machines) design things (i.e., to design process improvements).

In this paper, the framework, which relies on archival of design information into properly structured databases, on information retrieval, and semantic analysis, is presented. The proposed framework is more in line with methods employed by search engines, such as the one by Google.

The proposed framework is largely motivated by Yi's previous work [2]. In [2], indexing values of social tags in the context of an information retrieval (IR) model were assessed using a latent semantic indexing (LSI) method. Socially tagged resources were classified into ten Dewey Decimal Classification (DDC) main classes. Then social tags assigned to the resources were used to represent them in LSI.

FIGURE 1 High-level framework for applying AI to product design.



Similarities between resources were measured, and the aggregated similarities, according to the ten DDC main classes, were compared [2].

Similarly, this paper presents a high-level conceptual framework (vision), listed in Figure 1, for utilizing big data analytics to harvest repositories of known good designs for the purpose of assisting with product design. While the framework is generic, mechanical product design is considered. The framework assumes that, during the course of design projects, design information is captured in structured fashion using software (SW) such as the Ecosystem [3]. Project binders from past design projects are then archived in databases and made available to designers working on new design projects. These design containers were referred as e-design notebooks [3]. The AI system is trained so that the system can provide the best possible guiding information, for new product design, and sanitize the design decisions made.

The benefits associated with the proposed framework are multifold:

1. By comparing new design content against the guiding designs (reference), the fidelity of decisions related to the new design can be improved, as indicated above.
2. Through deployment of latent semantic analysis, the AI system can process a variety of user queries, retrieve the most relevant archived information, and present to the user.
   - In this paper, a simple example involving idea generation (brainstorming) for Concept Design is presented in order to provide insight into the significant utility that may be derived from the proposed framework.
3. While a relatively simple example is elected here, to convey the concept, more nuanced examples can be crafted around the Detailed Design phase.
   - Depending on users' needs, the AI system can query for information related to specific standards, regulations, policies, customer information, internal requirements, best practices, previous solutions, analogies, material properties, common components, etc., retrieve information from the databases yielding the best match, and present to the user.

The System Model

We model mechanical designs in terms of a single-layer neural network:

$$\tilde{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \ddots \\ \vdots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} = \tilde{A}\tilde{x}.$$

The input vector, $\tilde{x}$, could be considered as design variables (criteria) or customer requirements. The transformation matrix, $\tilde{A}$, could be a function of $\tilde{x}$ and $\tilde{y}$. Engineers transfer the requirements, $\tilde{x}$, into the product, $\tilde{y}$, through the transformation. The transformation matrix, $\tilde{A}$, may contain reasoning and knowledge to make $\tilde{y}$ (design $\tilde{y}$). In this paper, it is proposed that $\tilde{A}$ should consist of customers, requirements, systems and assembly. We apply artificial intelligence (AI) to train the system model.

The design criteria, $\tilde{x}$, could, for example, contain the desired weight, width, height and length of an automotive part. The elements of the product vector, $\tilde{y}$, could capture performance of the finalized part, or even ideas or options relevant to specific design stages.

For clarification, refer to the example below. It is assumed that the design organization has practiced structured capture of past design projects in SW like the Ecosystem [3, 4]. Design binders from these projects may, for example, have been archived in an internal database. For simplicity, it is assumed that input parameters take on values from a continuous range. It may be recognized that a straight forward application of a single-layer neural network model may not accommodate all requirements. To handle binary requirements (simple presence or absence), or XOR-like conditions, a two-layer neural network may be necessary [5].

The Design Project Binders

Design Process Assumed

It is assumed that a classical design process consists of Requirement Gathering, Concept Design, Detailed Design and Final Design. Such process is modeled in the Ecosystem SW [3, 4]. The customers, customer requirements and corresponding engineering requirements are defined, as a part of the Requirement Gathering, and captured in the Product Design Specification (PDS). The Concept Design consists of brainstorming, concept design analysis (scoring) and design selection. The Detailed Design may capture detailed analysis of both the overall system and associated subsystems. Final Design is usually preparation for prototype building or production, and may include steps such as testing and requirement validation [3, 4].

Archived Project Binders

The archived project binders in Figure 1 consist of past project binders, and are taken to represent known good designs. The project binders may contain pointers to pertinent content, based on designer inputs and available information. The input format captures and preserves content and associates with the relevant context. This facilitates storage for future use. Pertinent third-party data is accessed from databases with available context provided. The databases may be owned by the tool (Ecosystem) vendor, a customer or a third party. Designers ultimately choose to consider the information that is most relevant for any given design decision. This arrangement allows designers to leverage digital content management to make more informed design decisions without losing focus of the primary design challenge.

The information developed for the project binders in Figure 1 consists of pointers to the PDS and design objects. The PDS comprises of requirement objects, in programming context, and the design objects are comprised of component and assembly objects. Both can have hierarchy imposed. The design data itself is stored in mass outside the application.

Binders for New Design Projects

For new designs, designers could extract the design vector, **x**, from the new requirements, apply to a trained AI system, and get the guiding design, **y**, as an output. The guiding design, **y**, could be a reference (starting point) for design of the new product. Such reference may help improve the fidelity of design decisions. If design decisions cause the product to deviate significantly from the reference, **y**, explanations are likely necessary.

Practicality

Binders for new design projects are assumed to have the same structure as the binders from the past design projects (and to be archived as such). Note that regardless of which Product Lifecycle Management (PLM) system a design organization elects to use, the design data needs to be entered once. The Ecosystem provides capability for exporting design data into formatted project reports. So the design data does not need to be entered more than once. Content from the exported reports can be used in progress reports or project presentations. As long as design organizations make sure that each design project gets archived after completion, data management is expected to require relatively minor effort.

Structure of the Database Objects

In this study, database objects suitable for mechanical product design are defined along with their associated attributes. By defining the databases based on function, four databases with seemingly reasonable attributes are proposed. The database management overhead associated with the proposed architecture is expected to be minimal.

The Customer and Requirement Objects

Figures 2 and 3 present embodiments of customer and requirement objects from a database containing the PDS objects. Through the PDS, the designer builds up a collection of pointers to pertinent design information objects. It is of key importance to define proper attributes for the object pointers in the PDS database, and formulate metadata and leading indices accordingly. For the PDS database, the object pointers considered pertinent are listed in Table 1 and Table 2. The constraints in Table 2 may be binary and can be relatively easy to verify. The performance requirements typically involve binary thresholds, and are judged in accordance to design performance relative to the threshold. The objectives involve no thresholds, but rather provide optimization considerations for decisions.

The AI framework is capable of generating, managing, and presenting content with relevance to the design problem at hand in the databases available. It is assumed that, during the course of a design project, the database continues to grow. If design content is not readily available through a third-party or in-house, designers are apt to define it.

Figure 2 shows how the PDS object can be built using pointers to a database, for the purpose of being big data compatible. The requirements in Figure 2 refer to customer requirements, whereas in Figure 3 we are referring to engineering requirements.

TABLE 1 Attributes pertinent to the customer objects in the PDS database [4].

Attribute	Description
Name	Organization, Person, Entity
Type	Internal, External, Other
Importance	Low, Medium, High
Requirement	Key to Requirement database: Requirement[]

TABLE 2 Attributes pertinent to requirement objects in the PDS database [4].

Attribute	Description
Name	Descriptive name for the Requirement object
Owner	Key to customer database: Customer[*i*]
Importance	Low, Medium, High
Type	Constraint, Performance or Objective
Characteristic	Key to characteristics database: Characteristic[*j*]
Units	Key to units database: Units[*k*]
Threshold	Value for binary assessment

FIGURE 2 Example of customer database objects [4].

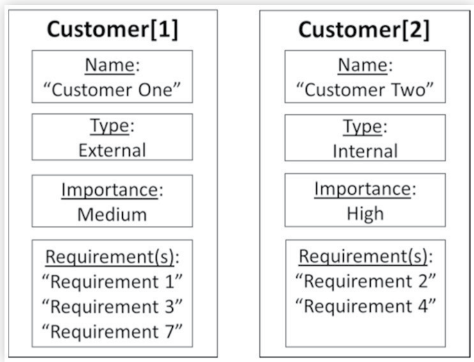
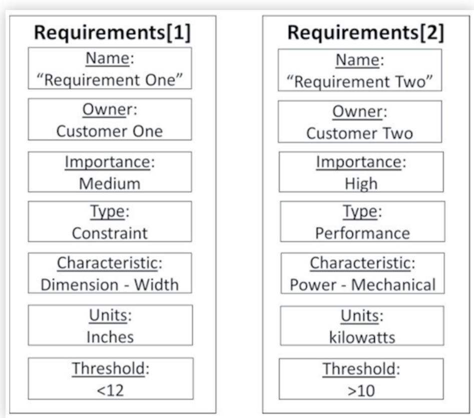


FIGURE 3 Example of a requirement database objects [4].



The Assembly and Component Objects

Figure 4 presents an embodiment of the assembly and component objects from the design database. The assembly objects consist of nested, aggregated subordinate levels, and have authority to define requirements applicable to the subordinates. The component objects consist of individual parts, pieces, or obtainable, self-contained assemblies. In case of the Design database, the pertinent attributes for the assembly and component objects are listed in Table 3 and Table 4. The rules

FIGURE 4 Illustration of a design database with assembly and component objects [4].

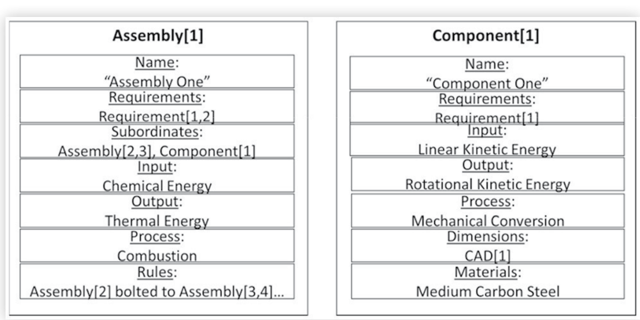


TABLE 3 Attributes pertinent to assembly objects in the design database [4].

Attribute	Description
Name	Descriptive name for the Assembly object
Requirements	Key to the PDS database: Requirement[]
Subordinates	Define subassemblies and components
Input	Key to database: Flow[]
Output	Key to database: Flow[]
Process	Key to database: Process[]
Rules	Key to Rules database: Rules[]

TABLE 4 Attributes pertinent to components objects in the design database [4].

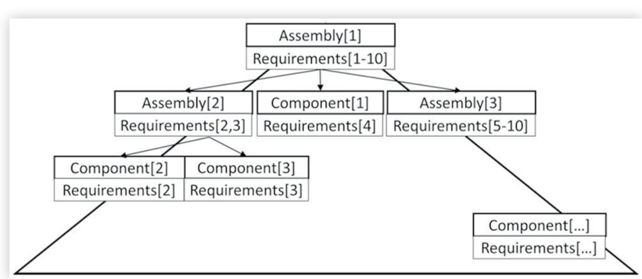
Attribute	Description
Name	Descriptive name for the Assembly object
Requirements	Key to the PDS database: Requirement[]
Input	Key to database: Flow[]
Output	Key to database: Flow[]
Process	Key to database: Process[]
Dimensions	Nominal and tolerance, in the form of solid model data
Material	Key to database: Material[]
Properties	Description of miscellaneous properties

in Table 3 specify the governing constraints of aggregated subassemblies and components. It is assumed that the design database complies with standard relational database (schema) formats for big data compatibility.

The Overall Design

Figure 5 shows how the component options and associated requirements, for an overall design (one comprising of multiple subsystems), can be programmed into the database, based on engineering knowledge gleaned from prior designs. This knowledge may, for example, be related to machine design text awareness of risks for certain components or uses.

FIGURE 5 Illustration of how component options and risks, for an overall design can be programmed into databases based on existing engineering knowledge (for example, machine design text awareness of risks for certain components or uses) [4].



Database Structures Supported

Big data analysis capabilities can be applied at organizations with design repositories arranged in the form of relational databases. The core concept of project binders accessing data in databases holding all the design information, and creating pointers to the pertinent design content, may be adapted to other database structures.

External Databases

As shown in Figure 1, the design content aggregated is multifaceted and covers a broad spectrum of inputs. It not only consists of the project binders from the designers, but also includes existing and previous design projects within an organization, plus the linked-in design files, the outputs from the design tools, material from the industry databases (results from context verification), the configuration scripts, examples, content of sample databases provided, legacy databases for known good designs, search analytics, information on manufacturing procedures, material characteristics, material prices, and parts that can be obtained from elsewhere, etc. The result is a sizable database of useable information available to designers and design organizations.

Data Annotation

Our intent is to present a classification system suitable for mechanical design. We think of the collection of archived project binders (e- design notebooks [3]) as books in a library. We assimilate the indexing of the project binders to cataloging of books in a library. And we compare the metadata defined for the project binders to the index labels placed on the book. Similar to the index labels helping with identification of book of interest, the metadata facilities rapidly processing and accurately responding to designer queries. We assume the design content gets tagged, in a similar fashion as Google tags all websites, to facilitate queries reflecting the users' needs.

Semantic Framework for Analyzing User Needs

In this paper, a user need is considered to occur as a mix of the following four elements: Function, cost, material, and energy. The first step in the analysis of project binders (e-design notebooks) is the automatic understanding of user needs. The tasks with user needs are:

1. To understand the statements of user need/ requirement.
   - Here we treat a user need as a query.
2. To retrieve previous design information, or examples, that yield a good match to the user need.

By doing so, the information retrieval framework proposed can provide design teams (workforces) with design information similar to the ones previously reported and harvested, for the purpose of enhancing design efficiency and efficacy. To deal with these tasks, we propose to adopt an indexing and retrieval method from the IR field, one referred to as Latent Semantic Analysis.

Latent Semantic Analysis

LSA is an extension of a classic IR model, the Salton's Vector Space model (VSM) [6]. LSA was developed as an information retrieval technique that discovers hidden semantic structure embedded in documents [7]. In more detail, complex relationships exist between words and surrounding contexts, such as phrases, statements or documents, in which the words are located. For the discovery of latent semantic relationships, LSA begins with the creation of a co-occurrence matrix, where the columns represent contexts and the rows represent words or terms. An entry (i, j) in the matrix corresponds to the weight of the word i appearing in the context j. The matrix is then analyzed by applying singular value decomposition (SVD) to derive the associated hidden semantic structures from the matrix. SVD is a way to factorize a rectangular matrix. For an m-by-n matrix, $\mathbf{A}$, with $m > n$, the singular value decomposition of the matrix $\mathbf{A}$ is the multiplication of three matrices: An m-by-r matrix $\mathbf{U}$, an r-by-r matrix $\mathbf{\Sigma}$, and the inverse of an n-by-r matrix $\mathbf{V}$, in that order. That is,

$$\mathrm{A} = \mathrm{U}\Sigma\mathrm{V}^T.$$

Here, $\mathbf{V}^T$ is the matrix transpose of $\mathbf{V}$, obtained by exchanging $\mathbf{V}$'s rows and columns. Then, $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns and $\mathbf{\Sigma}$ is a diagonal matrix. Such a multiplication form is referred to as the SVD of $\mathbf{A}$. The diagonal elements of $\mathbf{\Sigma}$ are all positive and ordered by decreasing magnitude. The original matrix, $\mathbf{A}$, can be approximated with a smaller matrix, $\mathbf{A}_K$, where $\mathbf{A}_K$ is obtained by keeping the first k largest diagonal elements of $\mathbf{\Sigma}$. By definition, k is the rank of the matrix $\mathbf{\Sigma}$. By applying SVD factorization to the matrix $\mathbf{A}$, context (e.g., a set of statements characterizing user needs) is represented in a much smaller dimension, k, rather than the original high dimension, m. Note that

$$k \leq n,$$

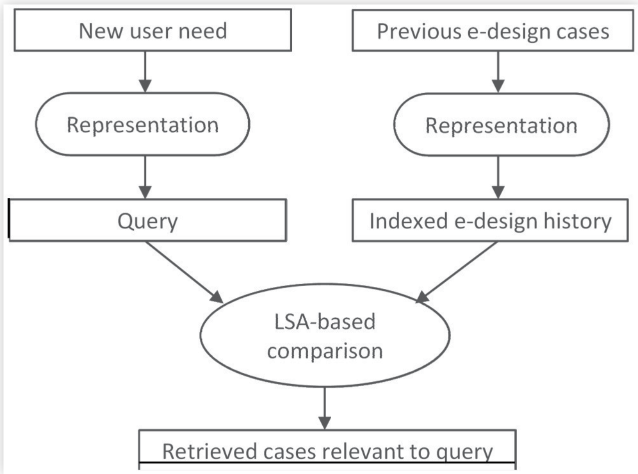
$$n \ll m.$$

As a result, a context is represented in a lower dimensional space, rather than in the full, much higher dimension. k is referred to as the dimension of the latent semantic structure of $\mathbf{A}$. A comprehensive overview of LSA can be found in [8].

LSA-Based Approach

The goal of the LSA-based approach proposed is to provide designers with access to previous design records that are relevant to the designers' need. In the vocabulary of Information Retrieving, a designer's need is equivalent to a query.

FIGURE 6 Illustration of the process of retrieving previous e-design history relevant to a user query.



LSA is adopted as the framework of retrieving designers' needs in this paper, because the method has been proven to be an effective ***unsupervised*** algorithm for IR [9]. In fact, we do not consider here any statistic ***supervised*** algorithm-based approach, such as neural networks or deep learning, since such an approach requires an enormous amount of previous e-design examples, or cases, that presently are unavailable; building a corpus of e-design history, including preceding cases or examples, is one of the goals for the next stage of this research.

Figure 6 depicts an LSA-based approach of retrieving e-design cases that are likely to satisfy a query (i.e., a user need). The LSA-based method predicts the degree of relevance of e-design examples to the query and presents the most relevant previous e-design cases, or examples, to the user.

Application of Big Data Analytics

The framework proposed assumes a holistic big data analysis and efficient utilization of a broad spectrum of available information. Through proper database representation of the design content, and references from the design project journals, one can categorize the data and run various cross-correlations (queries) across projects or within projects. By storing the comprehensive design history in a cloud, and harvesting repositories of known good designs through database queries, one can improve the design decision fidelity for new designs. Access to such repositories can also prove invaluable for the purpose of post-mortem failure analysis.

The AI network in Figure 1 can be trained, for example, using the Delta Rule, which is sometimes referred to as the Widrow and Hoff learning rule, or the least mean square (LMS) rule [10]:

$$\Delta w_{ijx} = -\varepsilon \frac{\delta E}{\delta w_{ij}} = \varepsilon \delta a_{ix},$$

where Δw_{ijx} represents the update applied to the weight at node (perceptron) between links i and j in a neural network [10]. E represents an error function over an entire set of training patterns (i.e., over one iteration, or epoch) [10]. ε is a learning rate applied to this gradient descent learning [10]. a_{ix} denotes actual activation for node x in output layer i [10].

Example

A simple example is presented here to show how AI can help with idea generation (brainstorming) in the Concept Design stage of a project involving the design of a reliable, single-operator Go Kart lift stand. This may be a capstone project, where the experience of the designers in the area may be somewhat limited. Therefore, they simply pose the following as input to the AI system:

"We need a reliable, single-operator stand for kart racers".

The system responds to the stated need by offering a number of ideas or options. Based on what can be retrieved from the databases, or the training data available, the system may offer the following suggestions:

"1. JEGS Multi-Purpose Lift
2. KartLift BigFoot
3. KartLift Winch Lift
4. Electric Super Lift
5. Go Kart Stand Lift".

Supplementing the overall process outlined in Figure 7, Figure 8 lists intermediate steps elucidating how the engine for latent semantic analysis is able to arrive at this conclusion.

While this example may come across as relatively simple (many mechanical designers may have a clue as to what type of lift stands are available), it conveys an application (illustrates the purpose) of the AI framework. More nuanced examples can be crafted, say, around specific standards, policies, material properties or common components. To our knowledge, there is presently no systematic search available for helping designers with brainstorming.

FIGURE 7 Illustration of the AI framework applied to the Concept Design stage of a project involving the design of a single-person Go Kart lift stand.

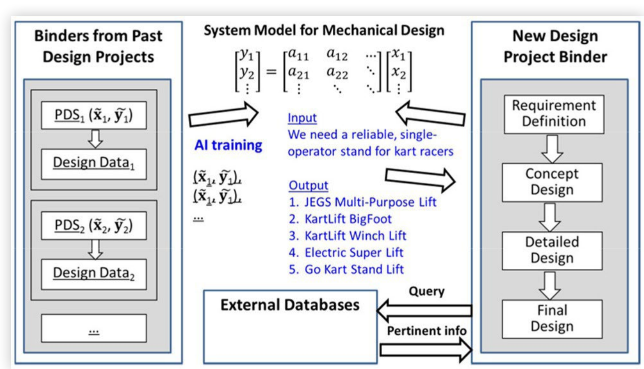
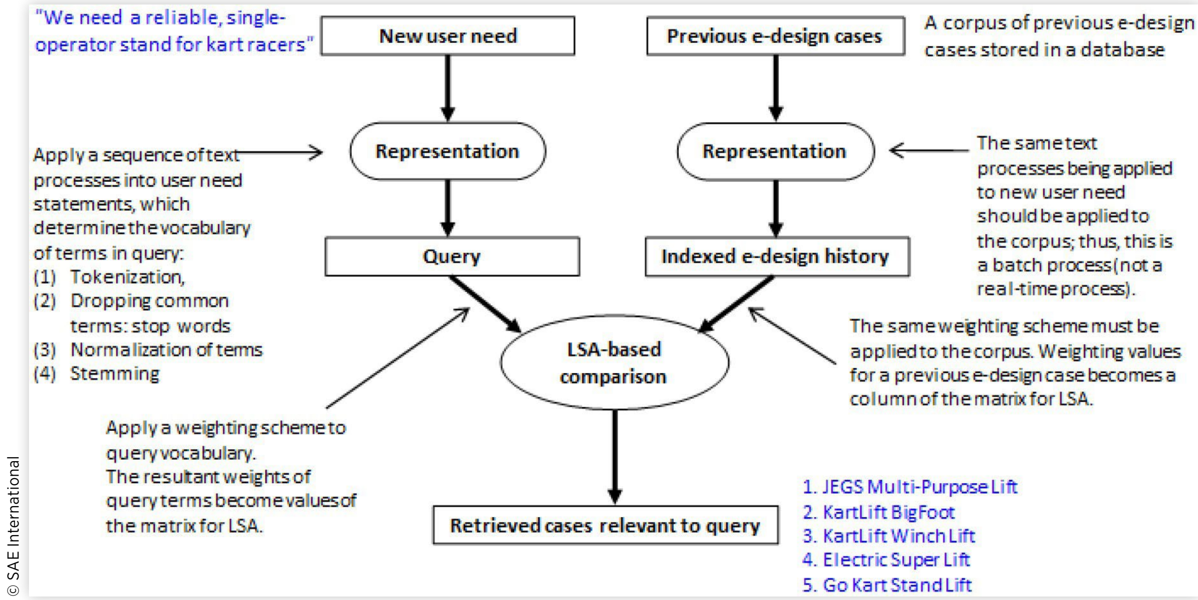


FIGURE 8 Latent semantic analysis applied to an example involving idea generation (brainstorming) for the Concept Design stage.



Important Questions Remaining

While this paper presents the high-level conceptual framework for utilizing AI for the purpose of aiding with mechanical product design, a number of important questions remain, such as:

1. How is the system going to validate the integrity of the archived data?
   - How are we going to prevent situations where low-fidelity input applied to the AI training produces low-fidelity output?
   - How exactly do we qualify "known good designs"?
2. What type of AI interfaces will the system be able to support?
   - Will the system be able to support both heterogeneous (distributed) and homogeneous architectures?

Towards Reasonable Answers

1. It is possible that low-quality data gets archived and that this may impact the decision making. The AI system should be able to offer good reference designs (**y**'s), but may not be able to provide exact answers for all cases. Yet, most search engines are based on similar concepts as the AI framework proposed (information retrieval). There are many ways to account for imperfectness in the archived data.
2. Similarly, large, distributed databases (enterprise applications), such as Apache Hadoop or Spark, can be supported through the API interfaces provided [11]. Hadoop provides a native Java API to support file system operations [12]. One can use WebHFDS to interact with the Apache Hadoop file system externally through a more user friendly REST API [13]. WebHDFS concept is based on HTTP operations like GET, PUT, POST and DELETE [13]. Operations like OPEN, GETFILESTATUS, LISTSTATUS are using HTTP GET, while others like CREATE, MKDIRS, RENAME, SETPERMISSIONS are relying on HTTP PUT [13]. The APPEND operation is based on HTTP POST, whereas DELETE is using HTTP DELETE [13]. Authentication can be based on user.name query parameter (as part of a HTTP query string). If security has been turned on, then the authentication relies on Kerberos [14].

Conclusions

This paper presents a high-level conceptual framework for the utilization of artificial intelligence (big data) for the purpose of aiding mechanical product design. This broad objective has been formulated in terms of an AI training problem. It is shown how a system model for a mechanical design can be trained, based on archived design projectbinders (so called known good designs). In addition, the paper illustrates how a guiding design, **y**, can be obtained by applying the design variables corresponding to a new design project, **x**, to a trained network. Design decisions for the new design can then be sanitized through comparison with the reference. It is demonstrated how a database for product design can be defined, based on function, and present seemingly reasonable attributes. While some important questions are yet to be fully addressed, the proposed AI framework is both practical and meaningful. By harnessing information retrieval and latent semantic analysis, the paper illustrates some of the significant potential of the AI framework, through a simple example addressing the idea generation (brainstorming) stage of Concept Design.

References

1. Brown, D.C., “Artificial Intelligence for Design Process Improvement,” . In: Clarkson J., Eckert C., editors. *Design Process Improvement*. (London, Springer, 2005), 158-173.
2. Yi, K., “A Study of Evaluating the Value of Social Tags as Indexing Terms,” . In: Chu S., Ritter W., and Hawamdeh S., editors. *Managing Knowledge for Global and Collaborative Innovations*. (Hackensack, NJ, World Scientific Publishing. Conference paper presented at the 6th International Conference on Knowledge Management, Hong Kong, December 3, 2009), 221-232.
3. Steingrimsson, B., Jones, R., Etesami, F., and Yi, S., “Ecosystem for Engineering Design - A Comparative Analysis,” *International Journal on Engineering Education* 33(5):1499-1512, 2017.
4. B. Steingrimsson and S. Yi, “Digital Ecosystem for Engineering Team Design,” U.S. Utility Patent Application 15,613,183, filed on June 3, 2017.
5. Duda, R.O., Hart, P.E., and Stork, D.G., *Pattern Classification* Second Edition (Wiley, 2001).
6. Salton, G., “A Vector Space Model for Automatic Indexing,” *Communications of the ACM* 18(11):613-620, 1975.
7. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R., “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science* 41(6):391, 1990.
8. Dumais, S.T., “Latent Semantic Analysis,” *Annual Review of Information Science and Technology* 38(1):188-230, 2004.
9. T. Landauer, D. McNamara, S. Dennis and W. Kintsch, *Handbook of Latent Semantic Analysis*, University of Colorado Institute of Cognitive Science Series, 2007.
10. Widrow, B. and Hoff, M.E., “Adaptive Switching Circuits,” . In: *IRE WESCON Convention Record*. Vol. 4. (New York, 1960), 96-104.
11. Spark API, “Spark Developer Documentation,” 2017, https://sparkplatform. com/docs/overview/api.
12. Big Data Zone, “Hadoop REST API - WebHDFS,” 2017, https://dzone. com/articles/hadoop-rest-api-webhdfs.
13. Hadoop, “WebHDFS REST API,” 2017, https://hadoop.apache.org/docs/r1.0.4/webhdfs.html.
14. Wikipedia, “Kerberos (Protocol),” 2017, https://en.wikipedia.org/wiki/Kerberos_(protocol).

Contact Information

Dr. Baldur Steingrimsson
Tel: 763-439-6905
baldur@pdx.edu

Acknowledgments

This research has been supported by the National Science Foundation Grants 1,447,395 and 1,632,408. The authors also want to thank Garth Eimers for reviewing early drafts of the paper and sharing valuable insights related to practicality and commercialization.

ISSN 0148-7191