Initial Development of the Engineering Genome Project–an Engineering Ontology with Multimedia Resources for Teaching and Learning Engineering Mechanics

Dr. Edward J. Berger, University of Virginia

Ed Berger is currently the Associate Dean for Undergraduate Programs in the School of Engineering and Applied Science at the University of Virginia. He is also Associate Professor in the Department of Mechanical and Aerospace Engineering. He teaches mostly sophomore mechanics courses.

The Engineering Genome Project – Developing an Ontology for Engineering Mechanics

Abstract

This paper reports on the development of an engineering mechanics ontology in support of the Engineering Genome Project (EGP). Ontologies contain both hierarchical (i.e., taxonomic) and relationship information, and they organize knowledge into highly-structured and expertly-constructed schemas. Ontologies can help facilitate the novice-to-expert transition for learners by making the implicit connections and relationships among complicated and cross-linked pieces of information (that experts know but novices do not) explicit. Our goal with the EGP is to empower learners to explore these detailed relationships among complicated concepts and topics in order to mediate the transition from novice to expert. This paper covers the initial development of the Engineering Genome itself – specifically, the development of the: (i) knowledge architecture and the classes in the ontology; (ii) the coarse-scale and fine-scale attributes used to describe the rich characteristics of the knowledge; and (iii) the relationship information describing the invisible-to-novices connections among disparate pieces of knowledge. We discuss key decisions we have made about the architecture of the knowledge, as well as the computer implementation of the ontology using the Web Ontology Language (OWL).

The paper will follow this structure. First we introduce the notion of controlled vocabularies and the overall conceptualization of the knowledge domain. Next, we illustrate the development of specific relationships and attributes, and highlight the challenges of establishing a knowledge architecture for seemingly straight-forward concepts (such as coordinate systems and units). Next we show how the ontology can also be linked to specific curricula and in particular learning outcomes associated with courses (as well as ABET) in which specific concepts are introduced. Finally, we explain and demonstrate the query procedures through which the ontology is mined for relationship information that–despite their expertise–experts may not fully be aware of. Our results so far indicate that an ontology can indeed be developed for engineering mechanics, and that the potential pedagogical uses for a carefully-constructed knowledge architecture are promising.

Introduction

"Genomics" is a term that has entered the common English lexicon in recent years, and its definition has evolved from its strictly scientific origins. Genomics originally referred to the study of the structure, function, and organization of the chromosomes of an organism, but has more recently taken on the meaning of studying the (relationships among) underlying building blocks of a system. For instance, we now speak of organizational DNA, which expresses the underlying strength and interactions of an organization's decision processes, information sharing, rewards systems, and management structures. Enterprise solutions for knowledge management present approaches to capture and archive an organization's institutional knowledge from its systems, employees, and partners, and to operationalize that knowledge in the organization's everyday practice (Shahnaqaz et al.[1], and the explosion of knowledge management journals in the past 10 years[2]). But a more accessible example of genomic thinking comes from pop culture: the Music Genome Project[3].

About ten years ago, a group of music performers, experts, and enthusiasts came together with the goal of creating "the most comprehensive analysis of music ever." They defined hundreds of

musical attributes ('genes') that contain the granular, essential information about a particular piece of music. They then set about categorizing individual songs according to the taxonomy they had developed. So the Music Genome is a collection of digital assets (the songs) tagged with highly granular, descriptive attributes (the genes), and organized into a searchable database. The result is Pandora–the Internet radio station that allows users to probe the Music Genome and create playlists based upon keyword searches. Pandora interrogates the Music Genome to create a playlist of songs that are genomically related – the songs are close neighbors in the genome. Songs are not arranged by genre, or by era, or by band geographic origin, or by sales, or by any other coarse metric or hierarchical relationship. Rather, songs in the playlist are presented based upon their genomic similarity, based upon shared attributes. The power of the Music Genome is that it exposes relationships among songs that, on the surface, appear to be unrelated (or perhaps only tangentially related). The listener instantly becomes educated about, and curious to explore, these relationships.

Now imagine that we create a similar structure not for music, but for the whole of engineering knowledge. This Engineering Genome would include multimedia learning objects, tagged with appropriate attributes and organized into a searchable database, and it would allow learners to interrogate its contents and explore the underlying relationships among the individual bits of engineering knowledge. When users interrogate the engineering genome, they will be presented with multimedia learning objects that are not organized by hierarchy, but rather based upon *genomic similarity* (along the lines of how Pandora matches songs genomically[4]. The EGP intentionally bridges traditional content silos (i.e., classes in the curriculum) and enables the learner to visualize and comprehend the elusive and subtle relationships among engineering concepts that, on the surface, appear to be unrelated (or perhaps only tangentially related). The Pandora metaphor is used in Table 1 to present a high-level view of the EGP.

The EGP defines a multimedia learning object as a sharable digital file containing information useful for teaching and learning, and encompasses a wide range of file types (examples: video and audio files in Quicktime or MP4, Matlab .m files, java applets, PDF files, etc.). The specifics of each multimedia asset depend upon context; we currently have files spanning a range of different teaching and learning tools, including: (i) lecture videos, (ii) video problem solutions, (iii) simulations/animations, (iv) Matlab .m files and other executables, (v) text-based resources in PDF. Many others are possible and the EGP can admit these and many other file types.

Learning "content" is, however, not enough; we want students to understand the relationships among seemingly disparate pieces of content. Since at least the 1970's, there have been persistent calls[5;6;7] for increasing synthesis and design in the engineering curriculum, for greater emphasis on deep inquiry, and for a general reversal of the compartmentalization of engineering content in the classroom. The ability to *integrate knowledge* is a key trait of the modern engineer, and traditional engineering curricula often struggle to instill this trait. The Engineering Genome, once fully realized, will address this critical need by building a cross-curricular tool that describes the incredible richness of relationships between pieces of "content", and therefore promotes student understanding and integration of knowledge.

Methods–Controlled Vocabularies

feature	Pandora, The Music Genome	The Engineering Genome
the digital assets (the "content")	individual songs	multimedia learning objects covering specific subjects in very detailed ways
the "genes"	rhythm, instrumentation, harmony, orchestration, lyrics, etc. (more than 400 in all)	mathematical model, solution technique, perceived difficulty, etc.
hierarchical relationships	very limited: band → album → song	more formal: branches and sub-branches of engineering knowledge, along with taxonomies of mathematics and other disciplines

TABLE 1. A high-level view of the EGP, using Pandora as a metaphor.

The core of the Engineering Genome ontology is the underlying taxonomy itself, which is informed by supporting taxonomies (math, physics) and the discipline taxonomy of engineering mechanics, and contains a rich set of categories and tags in a relational database. The hierarchical organization of the supporting taxonomies presents itself as a parent-child relationship with set of coarse-scale attributes for all topical areas with the fine-scale taxonomy varying as appropriate by discipline. A controlled vocabulary, a set of standard terms, will be used for each of the supporting and discipline taxonomies.

In traditional information systems (i.e., library catalog), controlled vocabularies help to bring together under a single word or phrase, all the material that is available on a particular topic. The main purpose is to provide some mechanism for querying multiple resources simultaneously and provide some commonality of description across the resources being made available for searching. Controlled vocabularies group similar objects together and ensure consistency for searching. The use of a predefined, authorized set of terms applied to objects by a domain expert improves the relevancy of a search result. Using terms from established taxonomies will allow the Genome to integrate with other systems or content (for example books and other items from a library catalog) if so desired.

The Knowledge Domain and Its Description
The Engineering Genome is both an expertly-constructed organization of engineering knowledge, as well as a library of multimedia learning objects that are categorized according to that organizational structure. As such, it takes an applied ontological view of the knowledge domain by emphasizing both hierarchical constructs (taxonomies) and relational features that do not necessarily follow parent-child lines. These relational features really do demand an ontological view rather than a strictly hierarchical one, and we will consistently use the term *ontology* to describe the Engineering Genome.

For the purposes of this research project, an ontology is composed of several parts: (i) a structured, explicit description of a knowledge domain that indicates relationships among classes of objects in that domain, (ii) properties or traits (which we will call "genes") that describe various attributes of those classes of objects, and (iii) individual instances (in our case, multimedia learning elements) that are classified according to the set of classes and traits from items (i) and (ii). As a concrete example, consider a multimedia file (say, a movie) that is a video solution to a

particular problem. In the ontology, this file is an "individual" whose position in the ontology (and relationship to other areas of the knowledge domain) is defined by the class(es) and traits that describe it. In natural language, we could describe this multimedia file (this "individual") as:

> the **object type** is a <u>video solution</u> from the **course** <u>Statics</u>; the **file format** is `.mov`; the **dynamic status** in the problem is <u>equilibrium</u>; it focuses on the **application area** <u>trusses</u>; the **equation type** <u>within the solution is algebraic</u>; the **coordinate system** is <u>2D Cartesian, non-rotating system with a fixed origin</u>; the **equation solution approach** is an <u>analytical method</u> from **mathematics domain** <u>linear algebra</u>; the **units** are <u>SI</u>; the **discretization approach** is to use <u>free body diagrams</u>; the **derivation approach** for the free body diagrams is the <u>method of sections</u>; the **learning outcomes targeted** are <u>ABET (a)</u> and <u>course learning outcome 1.c</u> as stated on the syllabus; and so forth...

where all the bold-faced words represent classes in the knowledge domain, and the underlined words represent traits and attributes of this individual. The Engineering Genome ontology captures and stores this information in a systematic, highly-structured way, enabling powerful search as described later in this paper. A very readable introduction to ontologies is presented by Noy and McGuinness[8].

To be sure, the Genome contains specific taxonomies to help organize information (example: a mathematics taxonomy[9]); but the ontological view expands the use of controlled vocabularies, exploits richer relationship information (beyond simple hierarchies), plugs into natural language processing tools and the semantic web, and generally represents a more sophisticated knowledge system. Moreover, well-constructed ontologies enable the use of "semantic reasoners", autonomous software tools that crawl through the ontology looking for logical inferences from the stated structure of the knowledge (for an overview, see[10]). Importantly, reasoners can detect (especially within large and complex ontologies) important logical relationships among elements of the ontology of which *even experts might not be aware*. The actual reasoning capabilities are not especially *functionally* important right now, but they are important insofar as the ontology must be developed carefully, with reasoning in mind, so that future reasoning tasks can generate insight about the underlying relationships in the knowledge framework.

The content experts who author the multimedia assets and commit them to the Genome are best positioned to define each assets position in the knowledge domain as defined by the ontology. As a result, each content expert will upload assets to the Genome and classify them according to the Genomes scheme. Each asset therefore represents one expert's view on how it is situated in the knowledge domain. It is possible that other experts might disagree on the details of the classification, and therefore we could consider some sort of Delphi-type process to more firmly establish a consensus among a group of experts, perhaps along the lines of6. However, given the volume of assets to be committed to the Genome (literally thousands of videos and other multimedia elements), it is not feasible to engage in a collaborative process for each one. The pathway right now allows these decisions to be made by the individual uploading the file to the Genome.

In addition to the highly-structured vocabulary used throughout the expertly-constructed portion of the Genome, a set of user-defined tags with comparatively less structure is also available. Users (including the experts who initially commit multimedia assets to the Genome) are able to input

tags that become attached to each asset. These tags can be highly specific, and are always defined by the user. Commonly-used tags will be available to choose from, or the user can create a new tag. This tagging functionality is essentially a required element of modern, web-based social tools, of which the Genome strives to be one.

To date, the Engineering Genome has focused on a specific sub-domain of engineering knowledge encompassing undergraduate engineering mechanics covered in courses such as Statics, Dynamics, and Strength of Materials. Implicitly, the Genome also includes the underlying mathematics associated with topics in these domains, especially calculus and differential equations, but also including linear algebra and other topics. Physics information also provides relevant foundational material to the Genome.

Using the parlance of Table 1, the Genome defines a series of coarse-level and successively finer levels of "genes" to describe each of the multimedia elements in its library. Coarse-level genes include the following:

- mathematical modeling: the features of the underlying mathematics, including equation type, boundary conditions, coordinate systems, and the like
- solution techniques: potential approaches to solving the mathematical problem, including analytical, numerical, or graphical techniques
- application domain: classification of the problem into the appropriate branch of engineering mechanics
- perception-based descriptors: subjective information (derived from both students and experts) on how "hard" or "easy" the problem is perceived to be, where it falls on the "theoretical" ⟷ "applied" spectrum, how time consuming it might be to solve, etc.

Underneath these course-level genes lies a multi-level sequence of finer-level genes that express, in ever more detail, the specific features and attributes of a particular problem.

Linkages to Local Curriculum and Learning Outcomes

In order to support the learner-centered intention of the Genome, it is important to include local curricular information into the underlying knowledge organization. For instance, local course names, course numbers, and specific course learning outcomes for our institution are embedded in the Genome. To date, we have done this for courses at our university, but it is clear that a large percentage of the underlying genes and information in the Genome will not be school-specific (i.e., it will be related to the broad knowledge domain of engineering). Nonetheless, the idea of including locally-relevant information for any institution using the Genome is important to make the Genome useful and relevant to students at many different locations. As described later in the section on **Query Processes**, students are able to search for material from a specific course at their institution.

Each multimedia element in the Genome, in addition to its classification according to the structure described above, can be linked to specific learning outcomes associated with a given course, or with ABET (a)-(k) or program-specific outcomes. As such, the learning outcomes form the set of genes that may or may not be expressed for a given problem. For instance, many engineering mechanics problems will strongly attach to ABET (a): ability to apply mathematics, science and engineering principles. Others will connect to ABET (e): ability to identify, formulate and solve engineering problems. In addition, each multimedia element also attaches to

specific course learning outcomes. A typical learning outcome from a sophomore mechanics course might be: "students will be able to construct correct free body diagrams and use them to solve statics problems". This set of ABET and local curriculum learning outcomes are captured in the Genome and form another part of the genetic profile of each multimedia element committed to the Genome.

We acknowledge that the course information, learning outcomes, and even ABET outcomes may shift over time, and this means that such information may need to be recoded for each asset in the Genome. This information is likely more time-evolving that the knowledge framework for engineering mechanics, which is relatively stable over time. While we have not confronted this challenge yet, two paths present themselves. First, we could certainly disallow searching on course numbers or specific learning or ABET outcomes when/if this information shifts substantially over time. This is non-optimal, but it is a technologically unsophisticated solution that is easily implementable. Second, and more useful, would be to develop a mapping from the "old" outcomes or course numbers or ABET outcomes to the "new" set of information, and automatically update the assets in the Genome with the mapped information. This is technologically more sophisticated but could be well worth the effort for students who wish to search on specific courses or outcomes. In either case, we do not expect the course or outcome information to shift significantly in the near-term, and we therefore have time to consider the appropriate solution to this important question.

Strength of Genomic Similarity
The key operation in the Genome is its searchability. And to facilitate this powerful searchability, another set of genes must be defined that express the genomic importance or relevance of a particular attribute for a particular multimedia asset. For instance, every problem in engineering mechanics has a coordinate system used in its solution. But there is a difference between a problem that *uses* a coordinate system for its solution, and a problem *about coordinate systems*. In dynamics in particular, there are many problems related to applying different kinds of translating or rotating coordinate systems, and it would be useful to be able to capture the importance or relevance of the coordinate system to the overall solution. Similarly, some problems might touch upon a particular learning outcome lightly, while other might be very targeted to that learning outcome. Given this reality, it is important to develop a way to capture these subtleties in how the multimedia assets relate to the underlying set of genes.

We have approached this problem as follows. The expression of a gene for a given multimedia file is rated on a 6-point Likert scale, with a rating of 1 being a weak affinity, and 5 being a very strong affinity. A rating of 0 means that the gene is not relevant at all for a given problem. So a garden-variety engineering mechanics problem that uses a fixed origin, 2D, Cartesian coordinate system would have attached to it a rating of 1 for this relevance gene. And a problem about using rotating coordinate systems to solve dynamics problems would have a rating of 5 for this relevance gene. For a problem that can be solved via several different approaches (say, a dynamics problem that can be solved using either an energy or a momentum approach), the relevance gene tied to the solution approach might rate a 1. Similarly, each of the ABET (a)-(k) learning outcomes would have a rating on this 0-5 scale indicating the degree to which the learning outcome is targeted by the multimedia asset.
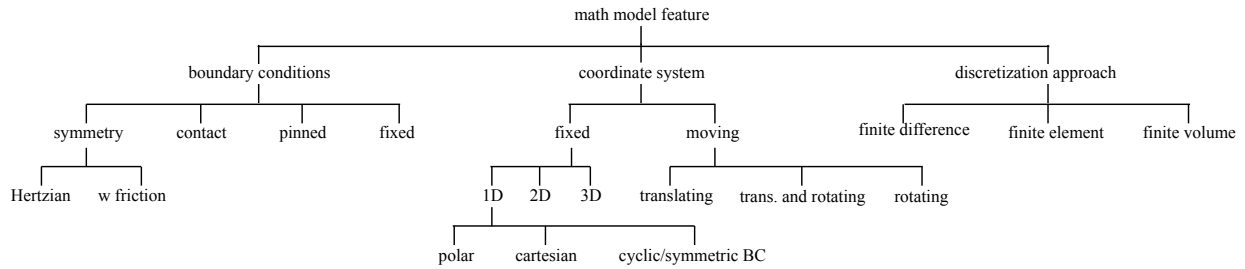
math model feature

boundary conditions   coordinate system   discretization approach

symmetry   contact   pinned   fixed   fixed   moving   finite difference   finite element   finite volume

Hertzian   w friction   1D   2D   3D   translating   trans. and rotating   rotating

polar   cartesian   cyclic/symmetric BC

FIGURE 1. An example (partial) mathematical modeling taxonomy showing detailed hierarchical relationships. This taxonomy is very incomplete and shown only to support the motivational discussion in the text.

This relevance scale is important for several reasons. First, it allows another layer of descriptive granularity to the genomic character of each multimedia asset. More granularity in the underlying description should correlate to more powerful search. Second, this relevance scale allows us to distinguish between multimedia assets which "use" something, versus those that are "about" that thing (as in the coordinate system example above). For experts, this is a completely transparent question, but *for novice learners this issue is critical*. It represents the differences in the specific focal areas for different problems, and answers the question: what exactly am I supposed to be learning about when I solve this problem? Finally, the relevance scale enables the different search algorithms we describe below, and in particular unleashes a very potent nearest-neighbor-type search that connects resources within the Genome that might otherwise seem unconnected.

Taxonomic Information Example. Consider a taxonomy of mathematical tools used to define engineering mechanics problems. We could define mathematics taxonomic information as shown in Figure 1. This taxonomy is very incomplete and is shown only to motivate the discussion presented here; the full taxonomy within the Genome is much more extensive and complete. Nonetheless, it is quite easy to appreciate how quickly the factorial combinations of traits from different depths of the parent-child hierarchy can add up, and the complexity of the classification scheme is revealed *to the experts who construct it*. Remember that the users are shielded from this complexity via a user-friendly front end (described in the section on **Software Architecture**).

This incomplete taxonomy also reveals one of the key challenges in organizing knowledge: there is not a single, unique, correct way to do it. Consider the branch of the taxonomy on coordinate systems. As presented, the first child-level classification answers the question of whether the coordinate system is fixed or moving. However, this taxonomy could just as easily have been constructed with the coordinate system dimensionality (1D, 2D, 3D) as the first-level child, with the question of fixed or moving relegated to a lower level of the hierarchy.

Ontological Information Example The ontological information comes into play when we consider specific multimedia assets that are classified according to the information structure and details. Each multimedia asset committed to the Genome is uploaded using a set of dialog boxes such as that shown in Figure 2, which is a screen shot of the portion of the current version of the Genome front-end used to link an asset to a set of ABET (a)-(k) outcomes. The uploader (who is a content expert and very likely the author of the multimedia asset) uses his/her judgment and knowledge of

ABET Outcomes

#	ABET outcome	N/A	1	2	3	4	5
A	An ability to apply knowledge of mathematics, science, and engineering	●	○	○	○	○	○
B	An ability to design and conduct experiments, as well as to analyze and interpret data	●	○	○	○	○	○
C	An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	●	○	○	○	○	○
D	An ability to function on multi–disciplinary teams	●	○	○	○	○	○
E	An ability to identify, formulate, and solve engineering problems	●	○	○	○	○	○
F	An understanding of professional and ethical responsibility	●	○	○	○	○	○
G	An ability to communicate effectively	●	○	○	○	○	○
H	The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context	●	○	○	○	○	○
I	A recognition of the need for, and an ability to engage in life–long learning	●	○	○	○	○	○
J	A knowledge of contemporary issues	●	○	○	○	○	○
K	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	●	○	○	○	○	○

FIGURE 2. A screen shot of a portion of the current Genome front-end showing how specific traits and their relevance within the class of ABET outcomes are defined for each multimedia asset uploaded.

the details of the asset to click on the appropriate radio buttons. There are similar dialog boxes for other classes of traits across the knowledge domain.

Software Architecture

The software architecture is split into two components: the front-end (which encompasses the user interface that masks the underlying complexity of the Genome beneath it) and the back-end (which manages the Genome's complexity and full taxonomic and ontological information). This configuration allows us to separate the part of the system that primarily deals with the management and display of the website (the front-end), from the part of the system that implements the knowledge organization and algorithmic searches (the back-end).

The back-end is implemented via a web service written in Java and running on an Apache Tomcat server. The back-end is primarily used to implement the more complicated algorithmic queries necessary for the Genome site; the remaining functionality is handled via the front-end. We chose a web service so that many different front-ends could all use the same back-end for these queries, and so that there would be no need to re-implement these aspects when porting the Genome system to a different platform. We felt that Java allowed for a more efficient means to program the algorithms necessary for the Genome. While there are many methods that the web service application programming interface (API, i.e., the way in which other applications can interact with the web service) provides, the primary ones are the methods that deal with the searching

FIGURE 3. The search page for the Genome; only the top part of this page is shown here.

routines. Like most web services, it is not called directly by the user, but via various web pages from the front-end.

The front-end is written in PHP, and uses the CakePHP framework[11]. Javascript is used on the client side for Web 2.0 functionality. The CakePHP framework includes all the model-view-controller (MVC) components of a modern RESTful web application. The front end is used for all of the website's functionality that is not handled by the back-end (i.e., all of the non-algorithmic functionality): display of the data, managing user privileges, validating user input, and performing the "simpler" data queries. The CakePHP framework allowed us to rapidly prototype the website, while also providing a solid foundation to build from.

Currently the functionality of the Engineering Genome includes:

- Allowing for the creation of categories, keywords, and a subject area hierarchy
- Defining various traits that each media file will have gene values for
- Allowing one to upload media files, and assign gene values for each trait; Figure 2 shows the selection of ABET category traits
- Allowing for searching of the uploaded media files; all searching is performed via a call to the web service, and the results are then displayed to the user

Query Processes
Queries are created by the end-user via the assistance of the front-end interface and sent to the back-end web service, where they are then parsed and returned as a list of multimedia assets deemed most relevant to the search by the back-end algorithms. As the user is not expected to know SQL, the front-end allows for an easy way to enter complicated queries. The search page is shown in Figure 3.

These queries, when parsed, are categorized as belonging to one of three categories:

- Keyword Searches
- Facet Searches
- Genomic Searches

These three search mechanisms, while currently implemented as standalone, independent searches, will eventually work in conjunction with each other to present a unified, multi-tiered search to the end-user. Users will enter a natural-language query into a text box, refine their results through the facet interface, and finally be presented with useful, genetically-relevant media files alongside their search results.

Keyword Searches. Keyword searches are built by the end-user with a traditional text box interface, as is used by the Google search engine. These query strings are then split into an array of words, which the back-end compares to the various classes, genes, and tags associated with each media file in the database, tabulating for each media asset a count of how many matches were made between its tags and subject areas and the words comprising the original query. Media files are then returned to the user in the order of their relevance to the keyword search. This is one of the simpler searching algorithms in the Genome, and provides an essential initial point of interaction between the user and the media files stored in the back-end database, from which the user can then refine and comprehend the results utilizing the other two searches. The very top of the keyword search box is shown in Figure 3.

Facet Searches. Facet searches are created through a series of drop-down and check-boxes, which provide delimiters to the user that allow specification of absolute conditions which must be met by media files that are returned by the web service back-end. Used in conjunction with the keyword searches, a user will be able to quickly refine and tailor search results. A typical facet-based search might focus on application area (the user is interested only in problems about trusses), or mathematical feature (the user is interested only in problems featuring a rotating coordinate system), or solution procedure (the user is interested only in problems involving numerical solution of differential equations). Query results from facet searches will be a subset of keyword searches. The facet search box is shown in Figure 3.

Genomic Searches. Perhaps the most powerful search approach, and the one that engages students in the novice-to-expert transition, is the search based upon genomic similarity. As we think toward the future when the Genome will contain a wide range of multimedia assets from different engineering disciplines, genomic search tools become a way to illustrate the connections between seemingly disparate and unrelated problems. The collection of traits describing assets in the genome spans an $n-$dimensional space, and this genomic search amounts to a high-order nearest-neighbor search along the lines of Pandora's search algorithm[4].

The output of this search is the set of multimedia assets that are genetically related; that is, they are related based upon some underlying features defined by the controlled vocabulary of the ontology. Genetic similarity is based not primarily upon a single gene or tag, as in keyword or facet searchers, but rather on a collection of genes that are, in the aggregate, expressed in each of the related multimedia files. Keyword and facet searches expose explicit similarities between multimedia assets (for instance, both keyword and facet searches will present multimedia assets in the same application area or that have the same underlying mathematics). However, the genomic search finds genetic near-neighbors wherever it can find them–including neighbors whose relationships are not explicitly clear. (As an aside, Pandora users know the learning that

takes place when you piece together how consecutive songs in a playlist are related.) In short, genomic searching enables discovery on the part of students.

An example: suppose a student starts with a keyword search on the term "truss". After viewing a multimedia asset related to trusses, the student performs a genomic search, seeking multimedia assets that are genetically similar to the truss example she just viewed. The student may be surprised to learn that truss problems and planar particle equilibrium problems are substantially, genetically similar because they share a *wide range* of genes (2D, fixed origin cartesian coordinate system, algebraic equations, linear algebra solution techniques, a method of joints-type discretization, and so on). This moment of discovery, of uncovering relationships that the student may have previously not known, supports the novice-expert transition and is the aspiration of the Genome.

Conclusions

The Engineering Genome currently under development presents promise as both an ontological tool for defining engineering mechanics domain knowledge, and learning tool that empowers students to search for and access learning materials. We are currently populating the Genome with engineering mechanics multimedia assets developed under a separate research program. The development so far has included careful construction of the knowledge domain, including specific "genes" for the relevant underlying features of the multimedia learning materials. Importantly, the inclusion of a relevance or importance gene allows us to clearly specify the most salient features of a particular multimedia element. The carefully-constructed information framework allows for powerful search following several algorithms. A brief demonstration of the Genome will be provided during the oral presentation of this paper.

REFERENCES

[1] M. Shahnaqaz, W. J. Doll, and X. Deng. A model of interrelationships among individual level knowledge management success measures. *International Journal of Knowledge Management*, 5(1):1 – 16, 2008.

[2] Knowledge management examples: *International Journal of Knowledge Management* (since 2005), *International Journal of Knowledge Management Studies* (since 2006), *Knowledge Management and E-Learning* (since 2009), *Electronic Journal of Knowledge Management* (since 2003).

[3] Pandora. http://www.pandora.com, December 2012.

[4] William T. Glaser, Timothy B. Westergren, Jeffrey P. Stearns, and Jonathan M Kraft. Comsumer item matching method and system. US Patent 7,003,616, February 2006.

[5] National Research Council. *Engineering Education and Practice in the United States: Foundations of Our Techno-Economic Future*. National Academy Press, 1985.

[6] National Academy of Engineering. *Educating the Engineer of 2020: Adapting Engineering Education to the New Century*. National Academies Press, 2005.

[7] J. Narum. Promising practices in undergraduate stem education. In *Evidence on Promising Practices in Undergraduate Science, Technology, Engineering, and Mathematics (STEM) Education Workshops*. Washington DC, 2008. accessed 7 January 2010.

[8] Natalya Noy and Deborah McGuinness. Ontology 101: A guide to creating your first ontology. Technical report, Stanford University Knowledge Systems Laboratory, March 2001. online, accessed 6 January 2013.

[9] National Mathematics Taxonomy Committee. Core subject taxonomy for mathematical sciences education. Technical report, http://people.uncw.edu/hermanr/MathTax/, May 2005. online, accessed 6 January 2013.

[10] John Davies, Dieter Fensel, and Frank van Harmelen, editors. *Towards the Semantic Web*. John Wiley & Sons, Ltd, 2003.

[11] CakePHP: the rapid development php framework. http://cakephp.org, December 2012.