

AIERO: An algorithm for identifying engineering relationships in ontologies



Paul Witherell^{a,*}, Ian R. Grosse^a, Sundar Krishnamurty^a, Jack C. Wileden^b

^a Department of Mechanical and Industrial Engineering, University of Massachusetts Amherst, 160 Governor's Drive, Amherst, MA 01003, United States

^b Department of Computer Science, University of Massachusetts Amherst, Amherst, MA 01003, United States

ARTICLE INFO

Article history:

Received 30 November 2011
Received in revised form 1 May 2013
Accepted 20 June 2013
Available online 20 August 2013

Keywords:

Semantic relatedness
Ontology
Product development
Change management
Interdependency
Consistency checking

ABSTRACT

Semantic technologies are playing an increasingly popular role as a means for advancing the capabilities of knowledge management systems. Among these advancements, researchers have successfully leveraged semantic technologies, and their accompanying techniques, to improve the representation and search capabilities of knowledge management systems. This paper introduces a further application of semantic techniques. We explore semantic relatedness as a means of facilitating the development of more "intelligent" engineering knowledge management systems. Using semantic relatedness quantifications to analyze and rank concept pairs, this novel approach exploits semantic relationships to help identify key engineering relationships, similar to those leveraged in change management systems, in product development processes. As part of this work, we review several different semantic relatedness techniques, including a meronomic technique recently introduced by the authors. We introduce an aggregate measure, termed "An Algorithm for Identifying Engineering Relationships in Ontologies," or AIERO, as a means to purposely quantify semantic relationships within product development frameworks. To assess its consistency and accuracy, AIERO is tested using three separate, independently developed ontologies. The results indicate AIERO is capable of returning consistent rankings of concept pairs across varying knowledge frameworks. A PCB (printed circuit board) case study then highlights AIERO's unique ability to leverage semantic relationships to systematically narrow where engineering interdependencies are likely to be found between various elements of product development processes.

Published by Elsevier Ltd.

1. Introduction

1.1. Motivation

Successful product development requires the concurrent and timely execution of many complex steps such as design, analysis, planning, and manufacturing. At each step, decisions are made, and initial decisions are frequently revisited [1]. Often, however, complexities in design spaces hinder the ability to maintain the needed level of consistency. These complexities can be managed by leveraging engineering relationships (interdependencies) within a product (and its development stages) to anticipate the impacts of decisions and modifications. Simulating these interdependencies in an engineering knowledge management system can help minimize information inconsistencies throughout a product development process [2]. The challenge, however, is that these interde-

pendencies, such as the one described below, are often not transparent.

Consider an automotive example provided by Mark Jennings of the Ford Motor Company [3]. When evaluating ways to improve a car's fuel economy, Jennings states that one approach is to reduce the load on the air conditioner by reducing the thermal mass of the seats. This rather shrouded trade-off scenario between cabin comfort and vehicle fuel economy is the result of an *interdependency* between the thermal mass of the interior seating and the cooling requirements of the air conditioner (AC). Here, the understanding is that decreasing the thermal mass of the seats reduces the loading requirements on the AC. Without possessing substantial experience and domain expertise, this trade-off would be difficult to identify. In multidisciplinary designs, and as products become more and more complex, the experience and expertise necessary to understand these complex engineering relationships is difficult to obtain.

Engineering interdependencies can also arise based exclusively on information-specific requirements, such as maintaining unit consistency or supporting model parameterization. While there are some knowledge management systems able to take advantage of interdependencies, such as change management systems, they

* Corresponding author at: National Institute of Standards and Technology, University of Massachusetts Amherst, Gaithersburg, MD 20899, United States
E-mail address: paul.witherell@nist.gov (P. Witherell).

are limited by the context and granularity of the information that can be generalized. These limitations make “out-of-the-box” advanced management applications difficult and create a necessity for customization [4], reinforcing the need for internal solutions. When discussing knowledge management in terms of change management, Rangan et al. [4] note, “The current brain trust in this area resides with the system integrators and PLM (Product Lifecycle Management) deployment specialists, and PLM deployment teams within companies [5,6].” In complex product development environments, especially those compounded by both geographic and multi-domain distributions, contributions from human expertise and experience can quickly be limited by cognitive abilities. Our research proposes that when identifying interdependencies, human expertise and experience can be supplemented with advanced knowledge management techniques.

In the context of a knowledge management system, an entity of information content (information artifact) can be considered “an entity that is generically dependent on some artifact and stands in relation of aboutness to some entity [7].” In the context of engineering, specifically product development, the artifact may “represent a distinct entity in a product whether that entity is a component, part, subassembly or assembly [8].” Properties inherent to information artifacts can be leveraged to identify interdependencies between their engineering entity counterparts.

Fig. 1 follows the transition of a radio amplifier from a physical object to a semantic representation at both the component and system levels. The transition from the physical layer to the virtual (information representation) layer depicts how physical objects can be modeled as entities of information, and that this information can be interconnected to create a complex system. This layer of information management is a representation of how product knowledge is currently managed, using database structures built on tables. The transition between the information representation layer to the semantic layer depicts how semantics provide structured relationships between entities of information. At the system level of semantics, it can be seen how these semantic relationships can quickly expand into a very complex system. We believe that the structured relationships created between semantic information artifacts can aid in the identification of engineering interdependencies in large and complex systems.

The presented work operates on the premise that semantic relationships between information artifacts in an ontological knowledge base can assist in the identification of engineering interdependencies. We propose that by using information artifacts as representations of their physical counterparts, semantics can be exploited to simplify some of the intricacies associated with the management of complex systems. Analytical methods can leverage

the strengths of semantic ties to narrow a scope of interest when identifying relationships in a knowledge base (Fig. 2). We can create a focus area with an increased likelihood of containing engineering relationships. This focus area can help domain experts identify interdependencies outside their domain expertise, or assist knowledge engineers by providing some insight into how domains interact.

1.2. Engineering knowledge management through ontologies

The potential contributions from much of the work presented in this paper rely on the engineering community continuing to adopt formal, structured knowledge management approaches. The use of ontologies, and the characteristics associated with them, are becoming a commonplace means for supporting product knowledge interoperability and life-cycle management [9–14]. In recent related works [2,15–20], the National Science Foundation Center for e-Design group at the University of Massachusetts Amherst has developed several Web-based modular ontologies for representing different aspects of the product development process, including design, engineering analysis, design optimization, and decision making. These ontologies and the resulting e-Design framework inspired much of the work presented in this paper.

1.3. Enhanced knowledge management through interdependencies

The authors have previously proposed methods to facilitate and guide portions of knowledge management in product development using Description Logic (DL) and Horn rules expressed in the Semantic Web [2]. DL ontologies were augmented with inference mechanisms to create a framework with the ability to identify conflicting knowledge and recognize the effects of changing information. The expressions of interdependencies in a product development process were shown to:

- (1) Enable corroboration of knowledge instantiations
- (2) Help maintain consistency during the knowledge instantiation process
- (3) Minimize redundancy in the knowledge instantiation process

To realize these advantages, the engineering interdependencies must first be identified. Modeling aspects of product development with ontologies creates unique “knowledge frameworks” where formal representations allow engineering concepts to assume ontological attributes. Similarities, or “likenesses,” between semantic representations of products and product development

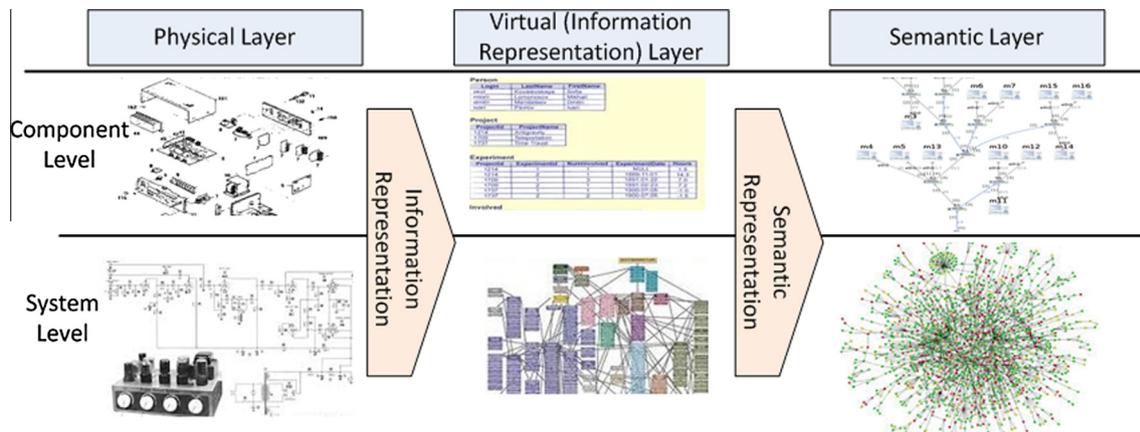


Fig. 1. Transitioning from physical to semantic interactions.

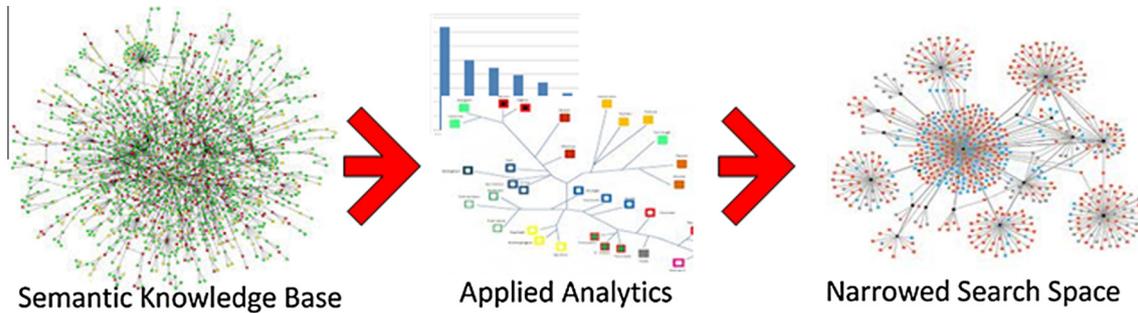


Fig. 2. Applied analytics to narrow search space.

processes can help identify interdependencies that can promote not only consistency, but also knowledge reuse and streamlined processes. Transitive associations made through “part-of” relationships can help identify interdependencies that may provide valuable insight into downstream implications of changes made within an integrated knowledge framework. To this end, this paper presents a novel semantic relatedness algorithm that quantifies ontological relationships to facilitate the identification of engineering interdependencies in the product development process.

2. Semantic relatedness

2.1. Overview

The term “semantic relatedness” refers to human judgments about the degree to which a given pair of concepts is related [21]. Semantic relatedness encompasses several types of lexical relationships, including synonymy, hyponymy/hypernymy, meronymy/holonymy, and antonymy. The hyponymy relation (i.e., “is-a” relation) is typically seen in a subsumption hierarchy (e.g., an ontology), and its inverse is known as hypernymy. Any relationship from the group of “component of”, “member of”, and “substance of” relationships are meronomic, and holonymic relationships are their inverses [22]. These relationships can help provide insight into not only how words, but also how concepts relate to each other.

2.2. Semantic relatedness techniques

Semantic relatedness measures categorize into four distinct categories: context vector, feature matching, path distance, and information content (IC) [23–27]. The following subsections discuss research in each of these measures while highlighting key techniques. The opportunity is also taken to address how well-suited each measure is for playing a potential role in the development of a new technique to help identify interdependencies within a product development framework.

2.2.1. Path distance

Path distance techniques are used to measure distances between nodes in “is-a” or “part of” hierarchies. The most basic of path distance techniques use summation of path lengths (which counts edges between concepts) to measure similarity between concepts [24].

Some variations of Rada’s approach have addressed differences in generality of subsumption relationships by scaling distance values based on the overall depth of taxonomies [25,28]. Other variations consider factors such as path direction and depth of the LCS (Least Common Subsumer). With respect to the product development process, “is-a” path distance measures possess relatively higher potential for inaccurate measurements due to the possibil-

ities of large variances between both the number of root classes and the depth of conjoined ontologies. Section 4.3 discusses the potential of “part-of” hierarchies further.

2.2.2. Information content and context vector

Generally speaking, path distance measures do not offer the most effective measurement of relatedness, especially when addressing large taxonomy structures. Discrepancies in measurements may be caused by variations in structures as well as inconsistencies due to concept generalities [26]. To address perceived limitations of path distance, Resnik [26] suggested that the similarity between concepts could be measured according to the frequency of an occurrence in a given corpus. The information content, or IC, of a concept c is calculated as:

$$IC(c) = -\log(freq(c)/(freq(root))) \quad (1)$$

where $freq(c)$ is the frequency of concept c and $freq(root)$ is the frequency of the root concept of the hierarchy. As the frequency of the concept increases, its IC value decreases with a lower limit of zero.

While IC measures have been widely adopted when calculating semantic relatedness, they are considered corpus-based and, therefore, require relating a large corpus of text to a general ontology such as WordNet [27,29]. Context vector measures share in this requirement [30]. Consequently, context vector and IC measures are better suited for use with lexical ontologies. They do not translate well to the domain ontologies necessary to represent the product development process. An arguably similar, alternative approach involves feature matching.

2.2.3. Feature matching

Cross has proposed [31] that path distance, IC, and feature matching are all very much related from the perspective of Tversky’s parameterized ratio model of similarity [32]. Tversky’s feature matching method, one of the more notable methods for identifying similarities, compares two concepts and expresses similarity as a ratio, calculated between 0 and 1, of their common and distinctive features:

$$Rel_{fea}(c_1, c_2) = \frac{|P_{c_1} \cap c_2^p|}{|P_{c_1} \cap c_2^p| + \alpha|P_{c_1} - P_{c_2}| + \beta|P_{c_2} - P_{c_1}|} \quad \alpha, \beta \geq 0 \quad (2)$$

where P_{c_1} and P_{c_2} are sets of features, or properties, belonging to two distinct concepts c_1 and c_2 , $(P_{c_1} \cap c_2^p)$ represents the set features shared by both concepts, $(P_{c_1} - P_{c_2})$ represents the features distinct to c_1 but not c_2 , and $(P_{c_2} - P_{c_1})$ represents the set of features distinct to c_2 and not c_1 . The values of these sets are reflective of their cardinality, shown by the vertical bars in Eq. (2). The scaling constants α and β can be used to specify the relative importance of each concept. A value of 1 is returned when two concepts are identical and a value of 0 is returned when two concepts do not share any properties.

In regards to identifying interdependencies in product development, feature matching is very pragmatic since it can help identify where values are most likely to co-exist between domains.

2.2.4. Combination techniques

Combinations of the four types of measures can exploit each measure's strengths. The ability to combine different measurement types affords semantic relatedness techniques the flexibility to be developed and tailored for specific applications. Many, if not most, have exploited this trait in implementations of semantic relatedness techniques.

2.3. Semantic relatedness applications

Notable projects involving the use of semantic relatedness techniques include the Human Genome biomedical ontologies and GIS (Geographical Information Systems) [33]. The Human Genome ontologies, including MeSH [23], SNOMED-CT [34], and ICD9-CM,¹ have used relatedness techniques to cross both language and geographical boundaries. In GIS, relatedness techniques have measured similarities between geographic features to support the identification of conceptually close but not identical objects. In the engineering community, Li et al. [35] have adopted relatedness techniques as a method for improving knowledge retrieval.

3. Semantic relatedness in domain ontologies

Despite numerous methods for measuring semantic relatedness, a common underpinning is that many, if not most, require a large corpus of text and the use of tools such as WordNet to implement them [21]. The transition of relatedness measures from lexical to domain ontologies has been initiated mostly through practice of ontology alignment [36], where relationships exist between concepts in lieu of words. When using similarity to establish correspondences between domain ontologies, sets of overlapping concepts, those which are similar in meaning but have different names or structure, must be determined [37].

Based on Rahm and Bernstein's work [38], Euzenat and Valtchev [39] have separated ontology alignment techniques into five distinct categories:

- (1) *Terminological comparison*: comparing the labels of entities.
- (2) *Internal structure comparison*: comparing domain attributes (e.g., the value range or cardinality of their attributes).
- (3) *External structure comparison*: comparing the relations of the entities with other entities.
- (4) *Extensional comparison*: comparing the known extension of entities, i.e., the set of other entities that are attached to them (e.g., instances).
- (5) *Semantic comparison*: comparing the interpretations (or more exactly the models) of the entities.

Applications of ontology alignment include agent communication, web service integration, ontology-driven data integration, and schema matching, among others [40].

4. Semantic relatedness in the product development process

The Algorithm for Identifying Engineering Relationships in Ontologies, or AIERO, is introduced here as a means to measure semantic relatedness and facilitate the identification of engineering interdependencies in an ontological product development framework. Of the technique categories outlined in Section 3, those

implemented by AIERO can be considered internal structure comparison techniques. AIERO exploits the ontology structure with a hybrid algorithm composed of the following three measures:

- (1) A feature-based measure that focuses on mapping concept properties. The more identical properties shared between two concepts, the higher their similarity. Feature comparison is useful for finding relationships that may lead to concepts sharing the same or similar information based on concept properties. When identifying interdependencies, this measurement helps identify those components that directly share information, such as relating two parts through an assembly, or two parameters through a model.
- (2) A second feature-based measure compares previously unmatched concept properties based on property ranges. Ranges can make properties similar without being equivalent, offering the ability to identify similarities at a more general level than feature comparisons. For instance, when identifying interdependencies, this measurement helps identify those components that indirectly share information, such as relating two parts through an assembly and subassembly, or two parameters through an analysis and an optimization model.
- (3) A novel meronomic measure is introduced into AIERO to quantify "part of" relationships. This measure operates on the principle that when a concept is a range of a second concept's property, that concept can be considered "part of" the second concept. The meronomic comparison provides a measure type uniquely appropriate for product development. The values of the properties used to define an engineering model will intuitively influence the definition of the model itself. This component helps identify interdependencies such as the influence an assumption made on a parameter may have on a model. The more one concept appears as a range of another concept, the higher the meronomic value.

The following sections review details associated with each component of AIERO, followed by their integration into a single hybrid algorithm.

4.1. Feature comparison component

The feature comparison component of AIERO is identical to Tversky's similarity measure from Eq. (2). Fig. 3 is a graphical representation of two concepts, c_1 and c_2 (shown as ellipses), being compared with feature comparison. Here, the intersecting set of features (shown as squares) ($P_{c_1} \cap P_{c_2}$) is equal to $\{p_a, p_b, p_c\}$, ($P_{c_1} - P_{c_2}$) is equal to $\{p_{d,c_1}, p_{e,c_1}\}$, and ($P_{c_2} - P_{c_1}$) is equal to $\{p_{f,c_2}\}$. While weighting can be useful, for demonstration purposes α and β will each be set equal to 1. Using the values given, the numerator, or $|P_{c_1} \cap P_{c_2}|$, is calculated as 3, equal to the cardinality of the set $\{p_a, p_b, p_c\}$. The two remaining values in the denominator

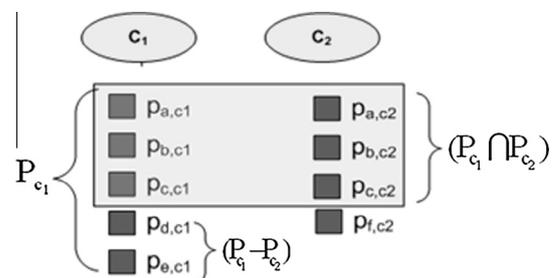


Fig. 3. Feature comparison between concepts c_1 and c_2 .

¹ International Classification of Diseases, 9th revision, Clinical Modification.

are found using the cardinality of the set $\{p_{d,c_1}, p_{e,c_1}\}$, equal to 2, and the cardinality of the set $\{p_{f,c_2}\}$, equal to 1. From Fig. 3, the relatedness value for AIERO's feature comparison component, or R_{fea} , can be calculated to be equal to:

$$R_{fea}(c_1, c_2) = 3/(3 + 2 + 1) = 0.5 \quad (3)$$

4.2. Range comparison component

In a product development framework, scenarios often exist where concepts may not be equivalent but can still be considered similar [41]. The second component of AIERO is an adaptation of the feature comparison approach used in the first component. In this component, the LCSs of range sets belonging to previously unmatched properties are compared. In an ontology, concepts, or classes, can be associated through the domains and ranges of properties. A property's domain specifies which concept(s) a property is an attribute of, and a property's range will bound the assumable values to a specific concept. Two properties may be considered similar when their bounded ranges belong to equivalent or similar concepts. The range comparison component of AIERO measures synonymy between concepts through the formal definitions (domains and ranges) of object-type properties as opposed to label equivalencies. This range comparison component is defined as follows:

- p'_{i,c_1} = Property p_i of concept c_1 that does not also describe c_2 .
- $P'_{c_1} = P_{c_1} - P_{c_2}$ = Set of all properties used to describe concept c_1 and not c_2 .
- r'_{k,p_i,c_j} = Range r_k used to describe property p'_{i,c_j} of concept c_j where $j \in \{1,2\}$
- R'_{p_i,c_j} = Set of ranges used to describe property p_{i,c_j} in concept c_j where $j \in \{1,2\}$.
- l'_{p_i,c_j} = LCS of set R'_{p_i,c_j} where $j \in \{1,2\}$.
- L'_{c_1} = Set of all LCSs of ranges of properties (l'_{p_i,c_1}) present in concept c_1 and not c_2 .

Thus:

$$p'_{i,c_1} \in P'_{c_1}$$

$$r'_{k,p_i,c_j} \in R'_{p_i,c_j}$$

$$l'_{p_i,c_j} \in L'_{c_j}$$

The AIERO component for comparison of ranges is then calculated using the following equation:

$$R_{int}(c_1, c_2) = \frac{|L'_{c_1} \cap L'_{c_2}|}{|L'_{c_1} \cap L'_{c_2}| + \alpha|L'_{c_1} - L'_{c_2}| + \beta|L'_{c_2} - L'_{c_1}|} \quad \alpha, \beta \geq 0 \quad (4)$$

Because an LCS is a generalization of the range of values a property may assume, not the direct property matching seen in AIERO's feature matching component, this component's contribution should be weighted less in the final, combination, algorithm. The reduction in weighting accounts for the fact that similar LCSs suggest, but do not guarantee, similar properties. The weighting is discussed further in Section 4.4.

Fig. 4 is a graphical representation of AIERO's range comparison component. Here, the ranges of the properties belonging to c_1 and c_2 that did not intersect in Fig. 3 are compared. Fig. 4 shows that L'_{c_1} is equivalent to $\{c_7, c_8\}$ and that L'_{c_2} is equivalent to $\{c_7\}$. From Fig. 4, and repeating the steps used in calculating the value of R_{fea} , the relatedness value for AIERO's range comparison component, or R_{int} , can be calculated as:

$$R_{int}(c_1, c_2) = 1/(1 + 1) = 0.5 \quad (5)$$

4.3. Meronomic component

In an earlier work, the authors developed a novel meronomic relatedness method [42]. A combination of edge counting and concept probability is used to determine how much an initial concept, c_1 , and its upper semantic cotopy, C_1 , is "part of" a second concept set C_2 , where C_2 is a set of only c_2 . A semantic cotopy consists of a concept and all concepts which subsume or are subsumed by that concept [43]. A value of 0 is returned if C_1 is not a part of C_2 , and a value of 1 is returned if C_1 is the only part of C_2 . The algorithm comparing a concept with itself, the value depends on how many other properties the concept has. Regardless, the argument can be made that objects are irreflexive, and, therefore, should not be compared with themselves at all [44].

In a meronomic tree such as that seen in Fig. 5, branches extend from a root concept set, such as C_2 , and each branch represents a property of which C_2 is a domain. In this figure, ellipses represent each concept and conjoining lines labeled "has part" represent concept properties. Each branch of the tree is extended through associations made by property domains and ranges. Nodes are added when one concept is a range of a property that has a domain of another concept. The subsumption of classes continues until any one of three criteria is met:

- (1) C_1 is subsumed by a branch from C_2 . Hence, C_1 is identified as being "part of" C_2 through that branch.
- (2) C_2 or a concept subsumed by C_2 is repeated in a single branch path, in which case to continue along the path would lead to redundancy.
- (3) C_2 or a descendent concept is not within a domain of any property, in which case the end of a branch has been reached.

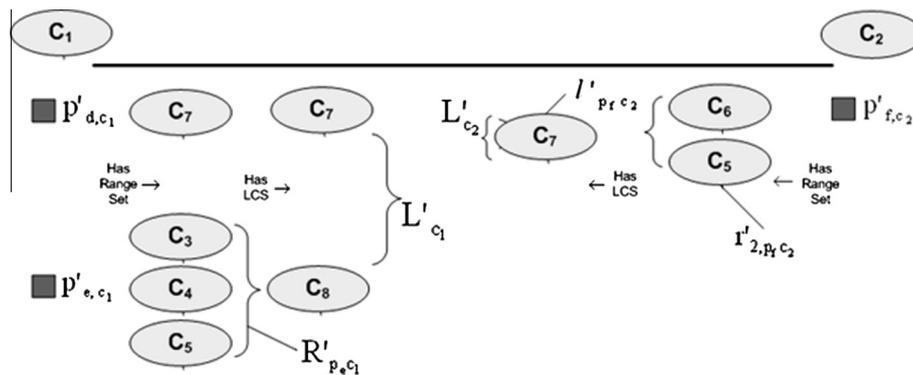


Fig. 4. Concept range comparison between concepts c_1 and c_2 .

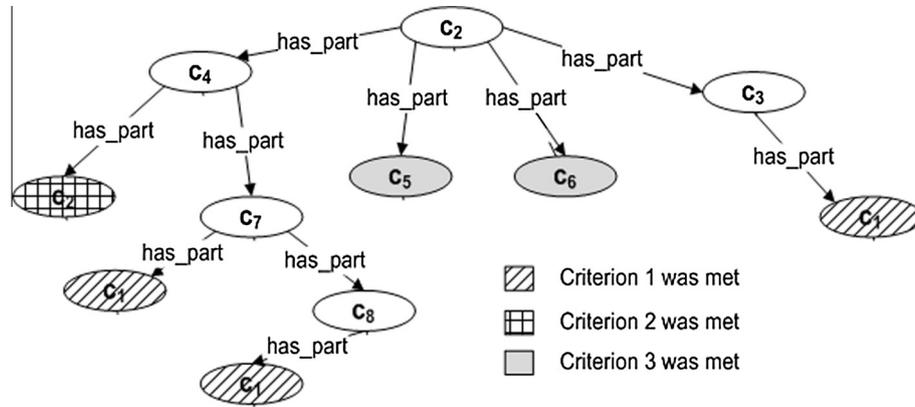


Fig. 5. Meronomic comparison to determine how much concept c_1 is “part of” c_2 .

The total relatedness value between two concepts can be calculated as seen in the following equation:

$$R_{\text{mer}}(c_1, c_2) = \frac{1}{B} \sum_{i=1}^{i=B} (Wt_{\text{branch}}(C_1, C_2)_i) \quad (6)$$

where B is the total number of branches protruding from concept C_2 and $Wt_{\text{branch}}(C_1, C_2)_i$ is the total contribution from each branch i . The total contribution from each branch is determined by the distance needed to reach a member of C_1 from the root concept C_2 . It is calculated by taking the product of the edge weights for each branch protruding from c_2 :

$$Wt_{\text{branch}}(C_1, C_2) = \prod_{C_i \in \text{path}(C_1, C_2)} wt(C_i, \text{parent}(C_i)) \quad (7)$$

where $wt(C_i, \text{parent}(C_i))$ represents the weight of each edge belonging to node C_i and its parent, $C_{i+1} = \text{parent}(C_i)$, along each branch. This approach reflects transitivity across the branch. It also allows for the relatedness contribution from each branch to be scaled based on the depth of the branch. The calculations to determine branch contributions are detailed in [42].

The first branch of Fig. 5 (far left), beginning with c_4 , expands to c_1 at two different levels. As c_1 is the only part of c_8 , its relatedness value contribution is 1. Although c_7 has two parts, they are both c_1 so the relatedness value is again 1. As c_4 has two branches, one leading to c_7 and one to c_2 , with only c_7 eventually leading to c_1 , the contribution from the branch associated with c_4 is 0.5. The second and third branches both terminate without a contribution, so the total contribution from each is zero. The fourth branch has only one path, and it leads to c_1 , so its contribution is 1. Therefore, the relatedness value of Fig. 5 can be calculated as:

$$R_{\text{mer}}(c_1, c_2) = \frac{1}{4} (.5 + 0 + 0 + 1) = 0.375 \quad (8)$$

4.4. Combination measurement

To help identify interdependencies in the product development process through semantic relationships, three separate measures have been introduced, one to address meronymy between concepts and two to address synonymy. The following metric combines the three measures presented into AIERO:

$$Rel_{\text{tot}}(c_1, c_2) = \alpha_m * R_{\text{mer}}(c_1, c_2) + \alpha_u * R_{\text{fea}}(c_1, c_2) + \alpha_n * R_{\text{int}}(c_1, c_2) \quad (9)$$

where c_1 and c_2 represent two concepts of a concept pair, and α_m , α_u , and α_n are weights for the meronomic relatedness term, the feature comparison term, and from the range comparison term, respec-

tively. It should be noted that each component of the combination relatedness measurement has been normalized. This is necessary due to the types of relatedness combined, specifically the combination of synonymy and meronymy.

The weighting factors in Eq. (10) can be altered to stress one type of measurement over another. Variations in desired values may be caused by such factors as differences in the comprehensiveness of ontologies (discussed in benchmark evaluations) or changes in the AIERO user's target objective. In general, the similarity measures are important to measure “likeness” between elements in product development, while the meronomic measure is integral to quantifying the “part of” associations that are most likely to reflect the propagating changes in a product development knowledge base. As noted in Section 4.2, because R_{int} is a generalization of R_{fea} , the recommendation is that R_{int} be weighted less when applying them concurrently.

5. Benchmark evaluations

The following benchmark scenarios evaluate the relative accuracy and consistency of AIERO across an ontology. The accuracy will be evaluated by studying AIERO's results and relating them back to intuitive assessments. The consistency will be evaluated by studying the results relative to others within the same ontology. Three evaluations of AIERO are performed independently within three separate ontologies. Ten different concept pairs are studied from each ontology to demonstrate the variations in values one might obtain within a single ontology. The three ontologies were deliberately chosen to represent three different levels of complexity. The three sets of ten concept pairs were deliberately chosen with the intention of creating contrasting results that could be intuitively understood and analyzed.

The first set of concept pairs is from a camera ontology from Pennsylvania State University [45], possessing a total of 27 classes and 8 object-type properties. The second set is from a suite of ontologies developed at the Technical University of Berlin for representing engineering artifacts [46], with 47 classes and 42 object-type properties. The third set is from the University of Massachusetts' e-Design ontological framework, comprising of multiple modular ontologies for modeling, analysis, and design optimization, with a total of 266 classes and 88 object-type properties.

For these evaluations, AIERO weights of α_m , α_u , and α_n were selected as 0.5, 0.3, and 0.2, respectively in Eq. (10). For those situations in which R_{int} was not applicable, α_m became 0.6 and α_u became 0.4. These weights were chosen to stress the relative importance of one concept being “part of” another when searching for engineering relationships, as well as the increased effectiveness

of feature matching over range matching when assessing synonymy. Concepts with a strong “part of” relationship, such as an assembly and its component, should return high marks. Alternatively, those concepts with little or no intuitive association, such as a material and person, should return comparatively low marks. Concepts that are similar, such as two components of an assembly, should fall in the middle. The results are shown in Tables 1–3.

The results from the camera ontology, seen in Table 1, are rather unrevealing, as six concept pairs returned values of 0.2. One mentionable irregularity is the identical scores of “memory card” to “film camera” and “memory card” to “digital camera,” as film cameras do not require memory cards. Such discrepancies will be attributed to the small scope of the ontology, as slight differences in the number of object-type properties used to define a class can lead to large discrepancies between classes. Conversely, the greater diversity of concept pairs from the engineering ontology led to more interpretable results (seen in Table 2). The concept pair of “weight requirement” and “requirement” returned the highest relatedness value, which is expected from their similarity contributions, as their definitions are very close. The next three highest concept pairs all came from the group of “engine,” “transmission,” and “powertrain.” These results are representative of the fact that all three are parts of a vehicle.

Table 1
Camera relatedness.

Concept 1	Concept 2	R_{fea}	R_{int}	R_{mer}	R_{rot}
Memory card	Battery	1.000	NA	0.500	0.700
Battery	Display	0.333	0.500	0.500	0.450
Camera	Manufacturer	0.000	0.000	0.875	0.438
Brand	Display	0.000	1.000	0.070	0.235
Memory card	Camera	0.250	0.000	0.250	0.200
Camera	Sensor	0.250	0.000	0.250	0.200
Sensor	Camera	0.250	0.000	0.250	0.200
Memory card	Film camera	0.250	0.000	0.250	0.200
Memory card	Digital camera	0.250	0.000	0.250	0.200
Display	Brand	0.000	1.000	0.000	0.200

Table 2
Engineering ontology relatedness.

Concept 1	Concept 2	R_{fea}	R_{int}	R_{mer}	R_{rot}
Weight requirement	Requirement	1.000	N/A	0.790	0.874
Engine	Transmission	0.714	1.000	0.900	0.864
Engine	Powertrain	0.400	0.333	0.906	0.640
Powertrain	Engine	0.400	0.333	0.899	0.636
Flange	Connector	1.000	N/A	0.333	0.600
Requirement	Flange	0.250	0.250	0.880	0.565
Engineering component	Transmission	0.667	0.000	0.600	0.500
Engineering component	Engine	0.667	0.000	0.600	0.500
Weight requirement	Powertrain	0.154	0.111	0.540	0.338
Test case	Flange	0.500	0.000	0.000	0.150

Table 3
E-design framework relatedness.

Concept 1	Concept 2	R_{fea}	R_{int}	R_{mer}	R_{rot}
Input parameter	Output parameter	1.000	N/A	0.167	0.500
Design model	Analysis model	0.875	0.000	0.097	0.311
Optimization model	Analysis model	0.824	0.000	0.113	0.304
Component	Assembly	0.455	0.000	0.313	0.293
Parameter	Constraint	0.333	0.000	0.333	0.267
Assembly	Component	0.455	0.000	0.150	0.211
Customer	Model	0.000	0.100	0.142	0.091
Material	Assumption	0.000	0.000	0.035	0.018
Projects	Units	0.000	0.000	0.000	0.000
Material	People	0.000	0.000	0.000	0.000

The e-Design framework represented the most diverse knowledge framework of the three, therefore returning the most contrasting, and revealing, results. As seen in the previous two tables, the highest concept pairs returned in Table 3 were again those that were most similar, such as “input” and “output” parameters. The relatedness between the concept pair “component” and “assembly” and the concept pair “parameter” and “constraint” also returned relatively high scores due to the meronomic relatedness between the concepts. Concept pairs “material” and “people” and “projects” and “units” returned expected scores of zero. It should also be noted that the individual scores were much lower overall due to the increased number of properties taken into consideration.

From a viability standpoint, the results of these assessments were encouraging, as AIERO’s results could be interpreted on a consistent basis across a single ontology. Consistency improved as the ontologies became larger and more complex; the relatedness values continued to diverge, and concept pairs were more distinguishable. The results, however, revealed a current limitation of this algorithm, namely that the measured values rely heavily on the comprehensiveness of the ontology from which the concept pairs are taken. To return unbiased results using the AIERO algorithm as presented, the development of one relationship type should not be heavily favored over the other. For instance by using only a single property to define a large set of concepts, a large synonymy value would be returned. To prevent bias between measurement types, the relationships being measured by AIERO, synonymy and meronomy, should exist at a similar frequency throughout the semantic framework. Alternatively, if there is a known bias, the weights used can be adjusted accordingly. The semantic framework should interlace the component and process knowledge just as the actual product would. A sound, homogenous semantic framework will curb AIERO from skewing results based on domain content. This idea will be discussed further in Section 7.

6. AIERO case study implementation

To evaluate AIERO’s effectiveness, results were related back to its main objective. The success of AIERO depends on its ability to facilitate the identification of interdependencies in the product development process. To this end, AIERO was applied to a knowledge framework for a printed circuit board (PCB) belonging to an audio tube amplifier.² This PCB case study provided a diverse set of parts and many possible complications, an effective scenario for evaluating the utility of AIERO. It should be noted that the selection of the 20 concepts, highlighted in the case study, was deliberate. For consistency, the weights used in this implementation will be the same as those used in benchmarking in Section 5.

6.1. AIERO implementation

The Audio Amp Framework (the e-Design framework specialized with an Audio Amp and PCB component ontology) was developed to test AIERO and its ability to narrow an ontology to a select set of classes to facilitate the identification of interdependencies. Current scalability limitations (see Section 7) restricted the set of classes to be evaluated to 20 (Table 4). Each class was compared with the other 19, leading to a total of 380 comparisons. The asymmetric nature of AIERO necessitated a comparison between each class pair twice.

To illustrate how AIERO calculated relatedness for each concept pair, the concept pair of “Idealization” and “Assumption” will be analyzed. This concept pair is intriguing, because, as defined within

² <http://www.dddac.de/>.

Table 4
Audio amp framework classes.

Class			
AUDIO:Audio Amp (Product)	EAMD:Load	MDKN:Input Parameter	ORGN:Projects
CMPT:Assembly	IDLZ:Assumptions	MDKN:Constraint	PCBCOMP:Physical Characteristics
CMPT:Company Developed	IDLZ:Idealization	MDKN:Objective Function	PCBCOMP:Operating Specifications
DSMD:Design	MATL:Material	MDKN:Output Parameter	PCBCOMP:PCB Components (Off theshelf)
EAMD:Analysis Model	MATL:Material_Behavioral_Model	MDKN:Units System	UNIT:Unit

the ontology, idealizations are founded on assumptions. In addition, each concept is defined by identical object-type properties. Therefore, these two concepts have traits associated with both synonymy and meronymy. Table 5 lists the object-type properties associated with each class, as well as the LCSs of these properties.

The identical properties shared by “Idealization” and “Assumptions” allowed for the following feature comparison:

$$R_{fea}(Assumptions, Idealization) = 5 / (5 + 0 + 0) = 1 \quad (10)$$

Because these two concepts share identical properties, their feature comparison resulted in a value of 1. This also eliminated the need for a range comparison, as the properties were equivalent, and therefore $R_{int}(Assumptions, Idealization) = 0$.

Although the two concepts share identical properties, the meronomic relatedness measurements between “Idealization” and “Assumption” and vice versa are asymmetric. When calculating how much an “Assumption” is part of an “Idealization,” as detailed in [42], the five properties denote there are contributions from five different “branches.” Similar to those in Fig. 5, these branches are a result of expanding meronomic relationships and creating a concept “tree.” Here, one branch leads directly to “Assumptions.” Two branches lead to “Idealization,” satisfying Criterion 2 as defined in Section 4.3, where the contribution from “Assumptions” is indirectly made through the “Idealization” node. After inserting the calculated values from each individual branch (Table 6) into Eq. (6), the total meronomic relatedness between “Assumptions” and “Idealization” was 0.49.

Using the simplified version of Eq. (10), the total relatedness between “Assumptions” and “Idealization” was as follows:

$$Rel_{tot}(Assumptions, Idealization) = 0.7(0.49) + 0.3(1) = 0.64 \quad (11)$$

Similar calculations were completed for each concept pair. Because AIERO is an asymmetric algorithm, it was initially necessary to compare concepts from two directions (a to b and b to a). For instance, although AIERO will yield a different value when computing the relatedness of the concept “Material” to the concept “Design” than when comparing the concept “Design” to the concept “Material,” an engineer will intuitively see the concept pairs as being equivalent, making the asymmetric duplication unnecessary. As asymmetry no longer needs to be considered during interpretation, duplicate pairs were eliminated and 190 concept pairs remained. Interpretations of the results of these calculations are discussed in the following sections.

Table 6
IDLZ:Assumptions “part of” IDLZ:Idealization.

IDLZ:Assumption “part of” IDLZ:Idealization		
Properties	LCS of Property	Meronomic Contribution
IDLZ:supports idealizations	IDLZ:Idealization	0.562
IDLZ:requires assumptions	IDLZ:Assumptions	1
IDLZ:requires idealizations	IDLZ:Idealization	0.562
MDKN:has creator	ORG:Person	0
MDKN:on model	MDKN:Model	0.325

6.2. Interpreting and utilizing AIERO results

As direct contributions from AIERO have ceased at this stage of the process (all relatedness values were calculated for the identified concept pairs), it becomes the responsibility of the domain expert to interpret the results. The interpretation begins with the creation of a focus area, comprised of the highest scoring concept pairs. Five of the 190 concept comparisons achieved an AIERO value of 0, indicating no interlacing between information artifacts, and these were subsequently filtered out. The comparisons of the remaining 185 concept pairs all resulted in AIERO values of some magnitude, and it becomes the responsibility of the domain expert to define the boundaries of the focus area before identifying the engineering interdependencies. The comparison of the concept pairs (EAMD:Load, ORGN:Project) and (IDLZ:Assumptions, MDKN:Input Parameter) will be discussed in detail to elaborate on the interpretation of AIERO results.

The concepts EAMD:Load and ORGN:Project (which represent a finite element load and an organizational project) intuitively have little in common, and therefore the AIERO results measuring the similarity of the two should be minimal. In fact, to analytically support this intuition, when using WordNet with Resnik’s similarity measure the calculated similarity is zero. To put this in perspective, the same measurement between two similar animals, a horse and donkey, was calculated to be 6.846, while the measurement between two distinctly different entities, a horse and a camera, returned a value of 0.059. A result of zero indicates that the two concepts are significantly different from one another.

In line with both intuition and Resnik, AIERO’s comparison of EAMD:Load and ORGN:Project returned values of 0 for both R_{fea} and R_{int} . However, The R_{mer} value between the concepts of EAMD:Load and ORGN:Project was calculated as 0.005, and R_{tot} was found to be 0.001. While this is a rather small number, the fact that AIERO

Table 5
IDLZ:Assumptions and IDLZ:Idealization classes.

IDLZ:Assumptions		IDLZ:Idealization	
Properties	LCS of Property	Properties	LCS of Property
IDLZ:supports idealizations	IDLZ:Idealization	IDLZ:supports idealizations	IDLZ:Idealization
IDLZ:requires assumptions	IDLZ:Assumptions	IDLZ:requires assumptions	IDLZ:Assumptions
IDLZ:requires idealizations	IDLZ:Idealization	IDLZ:requires idealizations	IDLZ:Idealization
MDKN:has creator	ORG:Person	MDKN:has creator	ORG:Person
MDKN:on model	MDKN:Model	MDKN:on model	MDKN:Model

returned a value greater than zero means the concept pair warranted some consideration. When expanded, it can be seen how a “load” propagates and contributes to defining a “project”; a load is used in a parameter, a parameter is used in a model, and a model is used in a project. As branches expand, contributions from EAMD:Load to ORGN:Project not only propagate, but also dissipate.

To benchmark the R_{tot} (EAMD:Load, ORGN:Project) result of 0.001, the comparison between the concepts IDLZ: Assumptions and MDKN:Input Parameter (representative of an assumption on a variable and an input parameter of a model, respectively) is discussed next. In the developed framework, assumptions are made on parameters, of which an input parameter is a type. Therefore, the expectation is that AIERO will return a relatively high value for the comparison of the concept pair of (IDLZ:Assumptions, MDKN:Input Parameter). This comparison should return a higher value than the previous, and this intuition is again supported by Resnik’s similarity measure, which returned a similarity value of 0.7794.

When comparing (IDLZ:Assumptions, MDKN:Input Parameter), AIERO returned an R_{fea} value of 0.083. This value was low because the properties compared were not exact matches. However, they did share many of the same concepts in their property ranges, as seen by the R_{int} of 0.571. The R_{mer} between (IDLZ:Assumptions, MDKN:Input Parameter) was 0.415, as assumptions were required in the definition of an input parameter. The total AIERO value, or R_{tot} , was found to be 0.364.

The R_{tot} value of 0.364 was large enough to rank the concept pair in the top ten percent of all concepts compared. Conversely, the 0.001 value returned by the (EAMD:Load, ORGN:Project) comparison placed the pair in the lower ten percent. While these results contrast starkly, a more complete picture is needed before deciding where to place a focus. Only after knowing results from all comparisons within a framework can the expert determine what “amount” of interaction establishes the cut-off point for identifying what pairs warrant additional focus. Many of the factors that influence the establishment of a “cut-off point” will be discussed in Section 7.

After considering and reviewing all concept comparisons evaluated for this case study, a significant separation was observed in AIERO’s results. Of the 185 remaining pairs, 89 of these pairs achieved scores of above 0.1 on the normalized scale. Using this rift as the cut-off point, the 76 below this score were removed from consideration. Though the possibility for an interdependency remains, it is important to remember AIERO was developed to improve efficiency and focus efforts when identifying interdependencies. With relatively minimal information exchange, these results were interpreted as belonging to concept pairs with very little or no possibility for the identification of an independence. At this juncture, the indirect contributions from AIERO have also ceased (AIERO’s values were used to achieve a focus area). It now becomes the responsibility of the domain expert to examine the remaining concept pairs and identify engineering interdependencies. This task was completed by the domain expert in [47] and will be discussed here for the sake of resolution.

Of the 89 remaining concept pairs, 25 of these included the classes “Unit” and “Unit System” due to their high meronomic tendencies (many of the domain concepts were associated with parameters, which used units in their definitions). While it is important to acknowledge the role these concepts play in instantiating knowledge, without any associated object-type properties, they had little impact in identifying interdependencies (other than consistency) in the analyzed framework. Subsequently, efforts were refocused on the 64 remaining concept pairs. From these 64, a total of 37, or 58 percent, of these resulted in the identification of interdependencies by the domain expert. The 77 relationships identified as a result from these comparisons ranged from the concept of parameters being shared between design and anal-

ysis models to the concept of a design model being part of a component [47].

7. Discussion

Semantics continue to play a role in advancing engineering knowledge management, providing a means to add structure and depth to information across systems and lifecycles. The algorithm outlined in this paper serves as a demonstration of how one might use semantic measurements, specifically semantic relatedness techniques, to help identify relationships between engineering artifacts. As noted in [4], defining relationships in engineering management systems requires expert insight into interactions between engineering artifacts. As system complexities continue to grow, the reach and effectiveness of domain experts will wane. Just as rules have been deployed in engineering management systems to reduce the number of human errors, new methods are needed to reduce the cognitive requirements of systems and domain experts when implementing new rules.

A design scenario for a printed circuit board was chosen as a proof-of-concept implementation of how semantic relatedness may apply to ontology-based engineering knowledge management systems. The PCB case study demonstrated how formal semantics can quantify relationships between information artifacts to facilitate the identification of engineering relationships. A focused group of concept pairs was created on the notion that the more two concepts interact semantically, as defined with AIERO, the greater the likelihood of an interdependency. Results from benchmarking support the proposal that semantic relatedness between engineering artifacts can be measured consistently across a comprehensive ontological framework. Results from the PCB case study support the notion that relatedness measures can help isolate concept pairs of interest across domains. High-value AIERO results normally indicated the existence of interdependencies, while concept pairs with lower returns intuitively had little or no correlation.

While the PCB case study highlighted many of AIERO’s potential contributions, it also exposed areas for potential improvement and further research. For instance, spreadsheet calculations were used in the implementation of the AIERO algorithm. While this was sufficient for a small sample size, a java-tool implementation with the ability to navigate an OWL (Web Ontology Language) knowledge framework is proposed as part of future work as a means to automate the algorithm’s execution. Beyond implementation strategy, the case study brought to light many other challenges that remain before a practical implementation of AIERO can be deployed.

The nature of the engineering concepts that can be addressed with our approach directly relates to how information artifacts are defined within the knowledge management system. If applications of an algorithm such as AIERO are considered during the development of a framework, and a framework is consistently developed with these applications in mind, the payoff can be significant. For instance, in OWL, this may include using more object-type properties when defining engineering artifacts, or using a deliberate subclassing scheme.

As AIERO results rely on the expansion and evaluation of semantic relationships, AIERO’s application creates an environment where insubstantial interactions will still result in a measurable value. The ability to return a value regardless of the amount of interaction is a necessary attribute of AIERO, as the structure of the ontology influences the magnitude of the normalized result values. This influence stresses the advantage of simultaneously making considerations for semantic applications while developing the ontological framework. This reliance on structure on a case-by-case basis also makes it difficult to assign a single cut-off value when determining relevant concept pairs. The

cut-off value is highly influenced by the structure of the ontology, its intended application, and the intent of the domain expert.

The indeterminate nature of the cut-off value also allows for the interpretation of what constitutes an “interdependency.” Given that there were no limitations on the “semantic distance” over which an interdependency can exist, the conceivable number of interdependencies within this case study is enormous. This, of course, was intentional as the algorithm is meant to assist the engineer, not replace him/her. However, for this reason, it is difficult to say when interdependencies “were or were not” identified. This introduces the notion of having context, intent, and perspective when identifying interdependencies.

Context is important to understanding what constitutes an interdependency and what does not, for instance a structural engineer may focus on mechanical properties while a fluids engineer may be more concerned with surface finish. The extent to which information artifacts are interdependent depends on the context of the domain and the application. This leads to intent. The specific interdependencies identified will depend on what knowledge is intended to be facilitated. Just because a potential relationship exists, and there is a high number of semantic interactions, does not necessarily mean it is an interdependency of importance. This brings us to perspective. Different experts will have different views on what constitutes a relevant interdependency, especially given the “distance” over which an interdependency can exist. For these reasons, we chose to focus here on the quantification abilities of AIERO.

Much work remains before conclusions can be drawn about AIERO’s overall effectiveness. Additional analyses must be executed across multiple ontologies and using multiple experts before the significance of the magnitude of the results can be discussed further. These analyses are needed to better understand how modifications in the AIERO algorithm affect the cut-off and move the “separation rift.” This is discussed in Future Work.

8. Summary

This manuscript introduces a novel approach to facilitate the identification of interdependencies in product development by leveraging ontological knowledge frameworks combined with semantic relatedness techniques. We outlined the details of AIERO’s development and presented a case study to examine the method’s applicability and usefulness. The results showed how the algorithm creates a focused area where there exists a higher level of domain interactions and therefore a higher likelihood of locating interdependencies. In the end, validation lies in either the discovery of interdependencies that may have otherwise gone overlooked or a decrease in the time or expertise necessary to identify interdependencies. Each of these are difficult to justifiably measure, even with significant repetition. Another form of validation comes in the ability to expand upon this work in future research.

AIERO is equally applicable to any ontology, independent of the implementation language. However, similar to concessions made by Li et al. [35], much of this work predicates on the assumption that ontologies will continue to be adopted by the engineering community. This research lays the foundation for continued work in the development of intelligent ontological knowledge frameworks, where the goal is to create an environment where implications of modifications to a distributed knowledge base are reflected in a consistent and productive manner.

9. Future work

Future work with the AIERO algorithm will focus on analyzing the results. While the case study addresses the application and,

to an extent, the utility of AIERO, it does not provide an in-depth analysis of the results. Further analysis of the effectiveness of AIERO will require identifying all conceivable interdependencies amongst the 20 concept pairs, from well-defined perspectives, to identify which interdependencies were overlooked. This will also provide further insight into how to define the separation criteria. Such an analysis is necessary to better understand the effect of varying the weights across the three measurements, and what impact this may have on successfully identifying interdependencies.

As noted earlier, interdependencies can very much depend on context, intent of the application, and the perspective of the individual. Therefore, further evaluation of the utility of AIERO’s results will require input from multiple experts from various ontologies. Additional evaluations will allow for further benchmarkings, against both other ontologies and other experts. These benchmarkings are necessary to understand to what extent the expert, context, and intent influence the utility of the algorithm.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 0332508.

References

- [1] O. Salomons, F. van Slooten, F. van Houten, H. Kals, A computer support tool for re-design, a prototype system resulting from applying a methodic design approach, in: *International Conference on Engineering Design*, 1993, pp. 1559–1570.
- [2] P. Witherell, S. Krishnamruty, I. Grose, J. Wileden, Improved knowledge management through first-order logic in engineering design ontologies, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 24 (2010) 245–257.
- [3] M. Jennings, Model driven vehicle system engineering: requirements, issues, and challenges, in: *ASME 2008 IDETC/CIE Conferences*, New York, NY, 2008.
- [4] R.M. Rangan, S.M. Rohde, R. Peak, B. Chadha, P. Bliznakov, Streamlining product lifecycle processes: a survey of product lifecycle management implementations, directions, and challenges, *Journal of Computing and Information Science in Engineering* 5 (2005) 227–237.
- [5] L.L. Bowler, S.M. Rhodes, A math-data environment to improve vehicle creation, in: *ASME 9th Annual Engineering Database Symposium*, Boston, MA, 1995.
- [6] Boeing, Boeing Production Initiatives, Define and Control Airplane Configuration (“DCAC/MRM”). <http://www.boeing.com/commercial/initiatives/dcacmrm/dcac_summary.html>.
- [7] NIF Standard Ontology, BioPortal, 2012.
- [8] S.J. Fenves, S. Fofou, C. Bock, S. Rachuri, N. Bouillon, R.D. Sriram, *CPM 2: A Revised Core Product Model for Representing Design Information*, National Institute of Standards and Technology (NIST), Gaithersburg, 2005.
- [9] S. Ahmed, S. Kim, K.M. Wallace, A methodology for creating ontologies for engineering design, *Journal of Computing and Information Science in Engineering* 7 (2007) 132–140.
- [10] M. Bajaj, C.J. Paredis, T. Rathnam, R. Peak, Federated product models for enabling simulation-based product lifecycle management, in: *ASME International Mechanical Engineering Congress and Exposition*, Orlando, FL, 2005.
- [11] K. Kim, H. Yang, D. Manley, Assembly design ontology for service-oriented design collaboration, *Computer-Aided Design & Applications* 3 (2006) 603–613.
- [12] G.M. Mocko, D.W. Rosen, F. Mistree, A critical analysis of description logic for engineering information management, in: *26th ASME IDETC/CIE Conferences*, Philadelphia, 2006.
- [13] J. Nanda, H. Thevenot, T. Simpson, R. Stone, M. Bohm, S. Shooter, Product family design knowledge representation, aggregation, reuse, and analysis, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 21 (2007) 173–192.
- [14] S. Szykman, R.D. Sriram, C. Bochenek, J. Racz, The NIST design repository project, in: R. Roy, T. Furuhashi, P. Chawdhry (Eds.), *Advances in Soft Computer-Engineering Design and Manufacturing*, Springer-Verlag, 1998, pp. 5–19.
- [15] R. Fernandes, S. Krishnamruty, I. Grosse, J. Wileden, Design and innovative methodologies in a semantic framework, in: *2007 ASME IDETC/CIE Conferences*, Las Vegas, NV, 2007.
- [16] R. Fernandes, S. Krishnamurty, I. Grosse, P. Witherell, J. Wileden, Semantic methods supporting engineering design innovation, *Advanced Engineering Informatics* 25 (2011) 185–192.

- [17] I. Grosse, J. Milton-Benoit, J. Wileden, Ontologies for supporting engineering analysis models, *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 19 (2005) 1–18.
- [18] J. Rockwell, I. Grosse, S. Krishnamurty, J. Wileden, A decision support ontology for collaborative decision making in engineering design, in: *The 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009)*, Baltimore, MD, 2009.
- [19] J. Rockwell, P. Witherell, R. Fernandes, I. Grosse, S. Krishnamurty, J. Wileden, A web-based environment for documentation and sharing of engineering design knowledge, in: *2008 ASME IDETC/CIE Conferences*, New York, 2008.
- [20] P. Witherell, S. Krishnamurty, I. Grosse, Ontologies for supporting engineering design optimization, *ASME Journal of Computing and Information Science in Engineering* 7 (2007) 141–150.
- [21] T. Pedersen, S. Pakhomov, S. Patwardhan, C. Chute, Measures of semantic similarity and relatedness in the biomedical domain, *Journal of Biomedical Informatics* 40 (2007) 288–299.
- [22] A. Budanitsky, G. Hirst, Evaluating WordNet-based measures of lexical semantic relatedness, *Computational Linguistics* 32 (2006) 13–47.
- [23] J. Backus, S. Davidson, R. Rada, Searching for patterns in the Mesh vocabulary, *Bulletin of the Medical Library Association* 75 (1987) 221–227.
- [24] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, *IEEE Transactions on Systems, Man and Cybernetics* (1989) 17–30.
- [25] C. Leacock, M. Chodorow, Combining local context and WordNet similarity for word sense identification, in: C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database*, MIT Press, 1998, pp. 305–332.
- [26] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, 1995, pp. 448–453.
- [27] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.
- [28] Z. Wu, M. Palmer, Verb semantics and lexical selection, in: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, Las Cruces, New Mexico, 1994, pp. 133–138.
- [29] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller, WordNet: an on-line lexical database, *International Journal of Lexicography* 3 (1990) 235–244.
- [30] S. Patwardhan, T. Pedersen, Using WordNet-based context vectors to estimate the semantic relatedness of concepts, in: *Proceedings of the EACL 2006 Workshop, Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, Trento, Italy, 2006, pp. 1–8.
- [31] V. Cross, Tversky's parameterized similarity ratio model: a basis for semantic relatedness, in: *Annual Meeting of the North American Fuzzy Information Processing Society*, vol. 63, 2006, pp. 541–546.
- [32] A. Tversky, Features of similarity, *Psychological Review* 84 (1977) 327–352.
- [33] F. Ferri, A. Formica, P. Grifoni, M. Rafanelli, Evaluating semantic similarity using GML in Geographic Information Systems, *On the Move to Meaningful Internet Systems: OTM Workshops*, Springer-Verlag, Cyprus, 2005, pp. 1009–1119.
- [34] C.o.A. Pathologists, fact sheet, in: *SNOMED-CT*, College of American Pathologists, 2004.
- [35] Z. Li, M. Yang, K. Ramani, A methodology for engineering ontology acquisition and validation, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23 (2009) 37–51 (Special Issue on Developing and Using Engineering Ontologies).
- [36] R. Albertoni, M. De Martino, Semantic similarity of ontology instances tailored on the application context, in: *2006 International Conference on Ontologies, Databases, and Applications of Semantics*, 2006, pp. 1020–1038.
- [37] N. Noy, M. Musen, PROMPT: algorithm and tool for automated ontology merging and alignment, in: *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, TX, 2000, pp. 450–455.
- [38] E. Rahm, P. Bernstein, A survey of approaches to automatic schema matching, *The VLDB Journal* 10 (2001) 334–350.
- [39] J. Euzenat, P. Valtchev, Similarity-based ontology alignment in OWL-Lite, in: L. Saitta (Ed.), *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04)*, IOS Press, 2004, pp. 333–337.
- [40] J. Euzenat, T. Le Bach, J. Barrasa, P. Bouquet, J. De Bo, R. Dieng, M. Ehrig, M. Hauswirth, State of the art on ontology alignment, *KnowledgeWeb* (2004).
- [41] P. Witherell, S. Krishnamurty, I. Grosse, J. Wileden, FIDOE: a framework for intelligent distributed ontologies in engineering, in: *2008 ASME IDETC/CIE Conferences*, New York, 2008.
- [42] P. Witherell, S. Krishnamurty, I. Grosse, J. Wileden, A meronomic relatedness measure for domain ontologies using concept probability and multiset theory, in: *2009 Annual Meeting of the North American Fuzzy Information Processing Society*, Cincinatti, 2009, pp. 338–343.
- [43] A. Maedche, S. Staab, Measuring similarity between ontologies, in: *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, Ontologies and the Semantic Web*, Springer-Verlag, 2002, pp. 251–263.
- [44] J. Patrick, Metonymic and holonymic roles and emergent properties in the SNOMED CT ontology, in: *Proceedings of the Second Australasian Workshop on Advances in Ontologies*, Australian Computer Society, Inc., Hobart, Australia, 2006, pp. 61–67.
- [45] J. Nanda, T. Simpson, S. Kumara, S. Shooter, A methodology for product family ontology development using formal concept analysis and Web ontology language, *Journal of Computing and Information Science in Engineering* 6 (2006) 103–113.
- [46] T. Tudorache, *Employing Ontologies for an Improved Development Process in Collaborative Engineering*, Technical University of Berlin, 2006.
- [47] P. Witherell, *Semantic Methods for Intelligent Distributed Design Environments*, Department of Mechanical and Industrial Engineering, University of Massachusetts Amherst, 2009.