Ontology-Based Knowledge Representation for Obsolescence Forecasting

Liyu Zheng
Virginia Polytechnic Institute and State University

Raymond Nelson III
University of Maryland, College Park

Peter Sandborn
University of Maryland, College Park

Janis P. Terpenny
Iowa State University, terpenny@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/imse_pubs

Part of the Systems Engineering Commons

The complete bibliographic information for this item can be found at http://lib.dr.iastate.edu/imse_pubs/20. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.
Ontology-Based Knowledge Representation for Obsolescence Forecasting

Abstract
Sustainment refers to all activities necessary to keep an existing system operational, continue to manufacture and field versions of the system that satisfy the original requirements, or manufacture and field revised versions of the system that satisfy evolving requirements [3].

The sales data is mainly in the form of number of units shipped. If it is not available, sales in market dollars or percentage market share may be used, as long as the total market does not increase appreciably over time [6].

For some products, within the same type of the product, life cycle curves characterized by parameters k, \( \mu \), and \( \sigma \) can vary with some primary attributes of the product. Examples are memory chips whose life cycle curves vary with different memory sizes. Memory size is the primary attribute describing the memory chip that evolves over time [6-8]. For these products, if the primary attributes of the product are not considered, the parameters k, \( \mu \), and \( \sigma \) obtained from the sales data of the product are only average values for that product.

The time range of the zone of obsolescence can be determined using data mining of historical data (e.g., last-order or last-ship dates) to achieve more accurate obsolescence forecasting [8].

Keywords
Center for e-Design, ontology, DMSMS, obsolescence, life cycle, forecast

Disciplines
Systems Engineering

Comments
This article is from *Journal of Computing and Information Science in Engineering* 13 (2012): 014501, doi:10.1115/1.4023003. Posted with permission.
Ontology-Based Knowledge Representation for Obsolescence Forecasting

Liyu Zheng
Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061

Raymond Nelson III
Department of Mechanical Engineering, University of Maryland, College Park, MD 20742

Janis Terpenny1
Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011
e-mail: terpenny@iastate.edu

Peter Sandborn
Department of Mechanical Engineering, University of Maryland, College Park, MD 20742

The impact and pervasiveness of diminishing manufacturing sources and material shortages (DMSMS) obsolescence are increasing due to rapidly advancing technologies which shorten the procurement lives of high-tech parts. For long field-life systems, this has led to an increasing disparity in the life cycle of parts as compared to the life cycle of the overall system. This disparity is challenging since obsolescence dates of parts are important to product life cycle planning. While proposed obsolescence forecasting methods have demonstrated some effectiveness, obsolescence management is a continuing challenge since current methods are very difficult to integrate with other tools and lack clear, complete, and consistent information representation. This paper presents an ontology framework to support the needs of knowledge representation for obsolescence forecasting. The formalized obsolescence forecasting method is suitable for products with a life cycle that can be represented with a Gaussian distribution. Classical product life cycle models can be represented using the logic of ontological constructs. The forecasted life cycle curve and zone of obsolescence are obtained by fitting sales data with the Gaussian distribution. Obsolescence is forecasted by executing semantic queries. The knowledge representation for obsolescence forecasting is realized using web ontology language (OWL) and semantic web rule language (SWRL) in the ontology editor Protégé-OWL. A flash memory example is included to demonstrate the obsolescence forecasting procedure. Discussion of future work is included with a focus on extending the ontology beyond the initial representation for obsolescence forecasting to a comprehensive knowledge representation scheme and management system that can facilitate information sharing and collaboration for obsolescence management. [DOI: 10.1115/1.4023003]

Keywords: ontology, DMSMS, obsolescence, life cycle, forecast

1 Introduction

Fast moving technologies have caused high-tech parts to have shortened procurement life cycles, rendering them obsolete quickly. This is referred to as DMSMS: the loss or impending loss of original manufacturers of items or suppliers of items or raw materials [1]. DMSMS obsolescence is a significant problem facing many long field-life sustainment-dominated systems. Sustainment-dominated systems such as aircraft avionics are often produced for many years and maintained for decades. The typical example is the B-52, a jet that was introduced in 1955, and is expected to be in service until 2040. For such long field-life systems, DMSMS obsolescence often emerges before systems are even fielded and recurs throughout their support life [2]. Obsolescence results in high life cycle costs for long field-life systems, and life cycle planning is essential for these systems to minimize the life cycle cost.

The obsolescence dates of parts are important information inputs during life cycle planning for long field-life sustainment-dominated systems suffering DMSMS obsolescence since there is rarely control over supply chains of these critical parts [3]. The obsolescence dates of parts can be obtained by forecasting. There have been several types of methods proposed including traditional methods such as ordinal scale based approaches, in which the life cycle stage of the product is determined from a combination of technological attributes [4,5], methods based on forecasting the product sales curve [6–8], and leading indicator methods, in which a leading indicator product can be further identified in each life cycle pattern of products that provides advanced indication of changes in demand trends of products [9]. Results have been mixed in the practical application of these obsolescence forecasting methods, where under certain situations, some of these methods can produce good forecasts and others not.

One of the greatest challenges is that obsolescence forecasting needs to work together with other methods such as design refresh planning for obsolescence management. However, tools for obsolescence forecasting are currently very difficult to integrate with other obsolescence management tools, because neither obsolescence forecasting tools nor obsolescence management tools are consistent in format. For example, the most prominent obsolescence management tool is the DMSMS knowledge sharing portal (KSP) by the Department of Defense [10]. It can report obsolescence status of parts, identify alternative and substitute parts, forecast part specific obsolescence risks, and consolidate part supply and demands. But the information within DMSMS KSP is inconsistent and cannot be integrated together. The common root cause of these tools is the lack of knowledge representation models for DMSMS obsolescence. In order to move efforts toward the goal of creating a web-centric enterprise-wide DMSMS management solution, ontology, a backbone information model, will be required [11]. Ontology is an explicit formal specification of the terms and their relations for sharing information in a domain [12]. There have been many application instances of ontology. In product design and development, an assembly design ontology has been developed for collaborative product development [13]. Ontology has also been applied to support product conceptual design [14]. Defined ontologies can be reused, although no specific ontol...
The Protege-OWL editor is an extension of Protege that supports the OWL. OWL is the most recent development in standard ontology languages, endorsed by World Wide Web Consortium (W3C) [15]. It is based on a logic model which makes it possible for concepts to be defined as well as described [16]. SWRL is an expressive OWL-based rule language that can be used to increase the amount of knowledge encoded in OWL ontologies. It is intended to be the rule language of the Semantic Web [17]. Finally, semantic queries can be executed based on SWRL rules for obsolescence forecasting.

In the sections that follow, we first consider several life cycle patterns along with their fit with common probability distributions. The framework of ontology representation for obsolescence management is then presented. The procedure of obsolescence forecasting and the life cycle ontology that represents life cycle stages of the product is described. An example based on a 64 Mbit monolithic flash memory is then provided to demonstrate the ontology representation for obsolescence forecasting. Finally, conclusions are provided.

2 Part Life Cycle Curve

The concept of product life cycle touches on nearly every facet of marketing and drives many elements of corporate strategy, finance and production. Product life cycle patterns (shown in Fig. 1) were first presented by Rink and Swan [18]. Although based on sales revenue, some of these are rarely seen in real life. As shown, Figures 1(b) and 1(c) bound off the bottom line, and (c) shows a repeating pattern, indicating products are revitalized after a niche market. Figures 1(e)–1(h) show the tendency of a stable pattern, i.e., products with life cycle patterns that have no DMSMS obsolescence. Figures 1(a), 1(i), and 1(j) show three of the most common types life cycle patterns: symmetrical increase and decrease in (a), fast increase versus slow decrease in (i), and slow increase versus fast decrease in (j). Products with these life cycle patterns will counter the problem of DMSMS obsolescence since sales drops below some limit of viability. (Figure 1(d) is simple and rarely seen and therefore not considered.)

The three most common product life cycle curves (a) (i) (j) in Fig. 1 can be obtained by fitting the sales data with the probability distribution function $f(x)$. Examples of possible probability distribution functions are listed in Table 1. The least squares method

<table>
<thead>
<tr>
<th>Name</th>
<th>$f(x)$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian distribution</td>
<td>$k e^{-\frac{(x-\mu)^2}{2\sigma^2}}$</td>
<td>$\mu$: location, $\sigma$: scale</td>
</tr>
<tr>
<td>Cauchy distribution</td>
<td>$\frac{1}{\pi(1+\frac{(x-x_0)^2}{r^2})}$</td>
<td>$x_0$: location, $r$: scale</td>
</tr>
<tr>
<td>Logistic distribution</td>
<td>$\frac{1}{1+e^{-\frac{x-x_0}{s}}}$</td>
<td>$x_0$: location, $s$: scale</td>
</tr>
</tbody>
</table>

Fig. 1 Product life cycle pattern [18]
Malcolmization with ontology in this paper. It is well suited since products such as these that can be represented with the Gaussian distribution life cycle are common. For example, the Gaussian distribution has been used by the Electronic Industries Association to define a standardized product life cycle [20]. As shown in Fig. 2, products with the Gaussian distribution life cycle pass through six stages: introduction, growth, maturity, saturation, decline, and phase-out. These stages take approximately the same time length. Life cycle patterns, like "and" and "or" cement simple class descriptions together to produce complex class descriptions [16]. The information about components and assemblies in a product is known, the other can be logically deduced by navigating the ontology. Examples are "has zone of obsolescence" and "is zone of obsolescence resolution of" relating the obsolescence forecasting class and the product class, and "has obsolescence resolution" and "is obsolescence resolution of" relating the obsolescence resolution class and the product class. Relationships between classes are summarized in Fig. 3.

Some important data properties of the product class related to obsolescence management are "sources," "physical characteristics," "interface," and "function." Sources represent organizations where products (parts) come from and include original manufacturers, distributors, brokers, and so on. According to the definition of obsolescence, the product (part) is obsolete if it can no longer be procured from original manufacturers. When obsolescence occurs (original manufacturers are lost), long field-life system sectors often must resort to obtaining parts from brokers and other aftermarket sources that may or may not be approved by the original manufacturer. For choosing sources, cost and obsolescence date also need to be taken into account. Physical characteristics indicate dimensional and material properties of a part that are generally measurable, such as form and fit; interface indicates an interaction relationship of the part with others in the system; and function indicates the role of the part plays in the system. Physical characteristics interface, and functions are important properties for seeking the substitution of the obsolete product (part). Only the substitutions with identical or similar slots can be used.

The framework of obsolescence ontologies is shown in Fig. 3.

When the product class is incorporated into the system as its equivalent part class, the built system can be described with description logic with restrictions and logic operators. A restriction describes a class based on the relationships that members of the class participate in. Typical restrictions are existential restrictions and universal restrictions. Existential restrictions, denoted by "some," describe classes of individual instances that participate in a least one relationship along a specified property to a specified class. Universal restrictions, denoted by "only," describe classes of individual instances that for a given property only have relationships along this property to a specified class. Logic operators like "and" and "or" cement simple class descriptions together to build up complex class descriptions [16].

The information about which parts build up the system can be obtained from the bill of materials (BOM). Object properties "has part" and "is part of" relate the part class with the system class. As shown in Fig. 4, system A consists of part a, part b, and part c. The closure axiom is applied in this example. The closure axiom makes the system "have and only have" those parts in the BOM. It includes the parts in the BOM with existential restrictions "some," and excludes the parts not in the BOM with universal restrictions "only."

### 4 Obsolescence Forecasting Ontology

Obsolescence forecasting formalized with ontology is based on product life cycle. In this section, the product life cycle class is described with six stages. Then obsolescence forecasting, realized with SWRL rules, is presented.
Table 2  Product life cycle stages and their characteristics [1,6,20]

<table>
<thead>
<tr>
<th>Stages</th>
<th>Sales</th>
<th>Price</th>
<th>Usage</th>
<th>Part modification</th>
<th>Competitors</th>
<th>Manufacturer profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Slow increase</td>
<td>Highest</td>
<td>Low</td>
<td>Frequent</td>
<td>Few</td>
<td>Low</td>
</tr>
<tr>
<td>Growth</td>
<td>Increasing rapidly</td>
<td>Declining</td>
<td>Increasing</td>
<td>Major</td>
<td>High</td>
<td>Increasing</td>
</tr>
<tr>
<td>Maturity</td>
<td>Stable</td>
<td>Stable</td>
<td>Stable</td>
<td>Periodic changes</td>
<td>Stable number</td>
<td>Stable</td>
</tr>
<tr>
<td>Saturation</td>
<td>Leveling out</td>
<td>Stable</td>
<td>Stable</td>
<td>Decline begins</td>
<td>Decline begins</td>
<td>Stable</td>
</tr>
<tr>
<td>Decline</td>
<td>Decreasing</td>
<td>Rising</td>
<td>Decreasing</td>
<td>Few or none</td>
<td>Declining</td>
<td>Reasonable for survivors</td>
</tr>
<tr>
<td>Phase-out</td>
<td>Lifetime buys offered</td>
<td>High</td>
<td>Decreasing</td>
<td>None</td>
<td>Declining</td>
<td>Reasonable for survivors (aftermarket)</td>
</tr>
</tbody>
</table>
4.1 Product Life Cycle Ontology. The product normally passes through six stages: introduction, growth, maturity, saturation, decline, and phase-out, as shown in Fig. 2. Each stage is associated with several characteristics such as sales, price, usage, and so on. Life cycle stages and their characteristics are summarized in Table 2. The product life cycle class is defined with data properties including “start,” “end,” and “characteristics.” Start and end describe start time and end time of the product life cycle period. And characteristics describe characteristics associated with the life cycle, which are shown in Table 2. A product life cycle class has six subclasses representing six stages of the life cycle. These subclasses inherit the properties from the product life cycle class. The start time of each stage class equals to the end time of its previous stage class if there is no special indication, except the start time of the introduction class and the end time of the phase-out class. Classes of the product life cycle ontology are shown in Fig. 5.

The covering axiom is applied to the product life cycle ontology. A covering axiom consists of two parts: the class that is being covered and the classes that form the covering. It specifies a class is covered by its subclasses, which means that a member of the class must be a member of at least one of its subclasses. If subclasses are disjoint, a member of the class must be a member of the only one of its subclasses. If subclasses are not disjoint, a member of the class must be a member of at least one of its subclasses.

\[ A \subseteq \bigcup_{i=1}^{n} B_i \]

Where

- \( A \) represents the product life cycle class;
- \( B_i \) represents the subclass \( i \) of the product life cycle class;
- \( B_1 \) represents introduction stage;
- \( B_2 \) represents growth stage;
- \( B_3 \) represents maturity stage;
- \( B_4 \) represents saturation stage;
- \( B_5 \) represents decline stage;
- \( B_6 \) represents phase-out stage.

The product life cycle ontology is applied with the covering axiom, and its subclasses are disjoint, so a member of the product life cycle class must be a member of either introduction, or growth, or maturity, or saturation, or decline, or phase-out, in accordance with the fact that a product can only be in one stage of the life cycle at a time. The description logic with restrictions and logic operators for the product life cycle class is shown in Fig. 6.

Not all products conform to the six life cycle stages. Some may undergo a false start and die out, and others may be revitalized after a niche market. The life cycle of these products may only consist of some of the six life cycle stages. These products are not discussed in the paper, but associated life cycle ontologies can be defined in the same way.

4.2 Obsolescence Forecasting With SWRL. For the product with the Gaussian distribution life cycle, the sales life cycle curve of a product is obtained by fitting the historical sales data with the Gaussian distribution using Eq. (1), that is

\[ \text{Minimize} \quad k, \mu, \sigma \quad \sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} \left( y_i - ke^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \right)^2 \]  

(3)

When the minimum of the sum of squared residuals is found, three parameters of the life cycle curve \( k, \mu, \sigma \) are determined. The time ranges of the zone of obsolescence and six stages are defined as following: the zone of obsolescence \( (\mu - 2\sigma, \mu + 2\sigma) \), introduction \( (\mu - 3\sigma, \mu + 2\sigma) \), growth \( (\mu - 2\sigma, \mu - \sigma) \), maturity \( (\mu - \sigma, \mu) \), saturation \( (\mu, \mu + \sigma) \), decline \( (\mu + \sigma, \mu + 2\sigma) \), and phase-out \( (\mu + 2\sigma, \mu + 3\sigma) \). The procedure of obsolescence forecasting for the product with the Gaussian distribution life cycle is shown in Fig. 7.

The procedure of obsolescence forecasting is represented with SWRL rules, and queries can be executed based on these rules. The SWRL rule is in the form of an implication between an antecedent and a consequent: antecedent \( \rightarrow \) consequent. Meaning, whenever the conditions specified in the antecedent hold (are true), then the conditions specified in the consequent must also hold (be true). An antecedent and a consequent are both conjunctions of atoms. Atoms are basic units that SWRL rules consist of, and can be of the form \( C(x, y) \), sameAs\((x, y)\) or differentFrom\((x, y)\), where \( C \) is a concept, \( P \) is a property, and \( x, y \) are either variables, individuals or data values [22]. The SWRL rule for obsolescence forecasting is shown in Fig. 8. A product is forecasted to be obsolete in the specific future time \( p \), if the start time of the zone of obsolescence is less than \( p \), where \( p \) is a constant in the execution which represents the time value.

This procedure of obsolescence forecasting is only appropriate for products with a sales life cycle that follows a Gaussian distribution. Other forecasting methods are required for products whose sales life cycle curve does not follow the Gaussian distribution. This class of problems is not addressed in this paper.

5 Example

This section demonstrates the ontology representation for product life cycle concepts and the method of obsolescence forecasting with a flash memory example. Flash memory is a common form of external memory that can be used by many electronic systems for storage and transfer of data. The recorded sales data of the 64 Mbit monolithic flash memory for the 1995–2002 time period provides that basis for the example (see Table 3). This memory was not introduced into the market until 1996.

The first step is to fit sales data with the Gaussian distribution function. When the objective of minimizing the sum of squared residuals is achieved, the life cycle curve of the 64 Mbit monolithic flash memory is obtained, as shown in Fig. 9 with a dashed line. The historical sales data is also shown in the same figure using a solid line. Since the data in this case is fitted well (dashed line), the curve line of historical sales data (solid line) is almost completely covered by the life cycle curve. The values of three parameters of the life cycle curve \( k, \mu, \) and \( \delta \) are \( 543.411 \times 10^6 \), 2003.861, and 2.0345 years. The time ranges of the six life cycle stages are introduction (1997.758, 1999.792), growth (1999.792, 2001.827), maturity (2001.827, 2003.861), saturation (2003.861, 2005.892), decline (2005.892, 2008.927), and phase-out (2008.927, 2011.961).

---

3The sales data is mainly in the form of number of units shipped. If it is not available, sales in market dollars or percentage market share may be used, as long as the total market does not increase appreciably over time [6].
4For some products, within the same type of the product, life cycle curves may be fitted well (dashed line), the curve line of historical sales data (solid line) is almost completely covered by the life cycle curve. The values of three parameters of the life cycle curve \( k, \mu, \) and \( \delta \) are different. For these products, if the primary attributes of the product are not considered, the parameters \( k, \mu, \) and \( \delta \) obtained from the sales data of the product are only average values for that product.
5The time range of the zone of obsolescence can be determined using data mining of historical data (e.g., last-order or last-ship dates) to achieve more accurate obsolescence forecasting [8].

To formalize obsolescence forecasting, related information of the product is represented with an ontological model, implemented using Protégé-OWL. Since the 64 Mbit monolithic flash memory is not further divided, it is an individual instance of a component class. The values of the properties related to obsolescence forecasting have been specified with data obtained from previous calculations. Figure 10 shows screenshots of the 64 Mbit monolithic flash memory ontology “monolithic_flash_memory_64M” example in Protégé-OWL. It has two important properties “has_life_cycle” and “has_zone_of_obsolescence.” Values of the properties point to the product life cycle “product_life_cycle_monlithic_flash_memory” and the zone of obsolescence “zone_of_obsolescence_monlithic_flash_memory” of 64 Mbit monolithic flash memory.

Based on the ontology representation for the product, queries for obsolescence forecasting can then be executed. For the 64 Mbit monolithic flash memory, a forecast is performed to determine whether or not it will be obsolete in 2010. As shown, 2010 is the specific future time of interest in the SWRL rule for obsolescence forecasting in Fig. 8. The class “Forecasted_to_be_obsolete_in_2010” is defined to contain all individual instances the product which are forecasted to be obsolete in 2010. After executing queries, the 64 Mbit monolithic flash memory appears in the class “Forecasted_to_be_obsolete_in_2010,” so it is forecasted to be obsolete in 2010. The process of obsolescence forecasting is completed. Figure 11 shows the screenshot of the SWRL rule and the result after executing queries in Protégé-OWL. In this example, the 64 Mbit monolithic flash memory is the only product which is forecasted to be obsolete in 2010.

6 Summary and Conclusions

This paper has presented an ontology framework for obsolescence forecasting. Obsolescence forecasting is a necessary part of DMSMS obsolescence management for long field-life sustainment-dominated systems. Information from the obsolescence forecasting (e.g., the obsolescence date of the part) is important input for design refresh planning (one way of strategic obsolescence management) of the systems. The obsolescence forecasting

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit shipments (millions)</td>
<td>N/A</td>
<td>0.1</td>
<td>12</td>
<td>35</td>
<td>90</td>
<td>200</td>
<td>360</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7 Flow diagram of obsolescence forecasting for product with Gaussian distribution life cycle

Fig. 8 SWRL rule for obsolescence forecasting
### Table 4  Forecasted sales data of the 64 Mbit monolithic flash memory after 2002

<table>
<thead>
<tr>
<th>Year</th>
<th>Recorded sales data</th>
<th>Life cycle curve</th>
<th>Residual</th>
<th>Squared residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996</td>
<td>0.1</td>
<td>0.31113913</td>
<td>-0.21113</td>
<td>0.044578</td>
</tr>
<tr>
<td>1997</td>
<td>1</td>
<td>1.84221418</td>
<td>-0.84221</td>
<td>0.709325</td>
</tr>
<tr>
<td>1998</td>
<td>12</td>
<td>8.566537579</td>
<td>3.433462</td>
<td>11.78866</td>
</tr>
<tr>
<td>1999</td>
<td>35</td>
<td>31.28549001</td>
<td>3.71451</td>
<td>13.79758</td>
</tr>
<tr>
<td>2000</td>
<td>90</td>
<td>89.73317458</td>
<td>0.266825</td>
<td>0.071196</td>
</tr>
<tr>
<td>2001</td>
<td>200</td>
<td>202.132216</td>
<td>-2.13222</td>
<td>4.546345</td>
</tr>
<tr>
<td>2002</td>
<td>360</td>
<td>357.5942078</td>
<td>2.405792</td>
<td>5.787836</td>
</tr>
<tr>
<td>2003</td>
<td></td>
<td>496.8414811</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td></td>
<td>542.1477424</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td></td>
<td>464.6114338</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td></td>
<td>312.7048136</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td></td>
<td>165.2919388</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
<td>68.6149912</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td>22.37191647</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td>5.728456208</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td></td>
<td>1.151978674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td></td>
<td>0.181938125</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Forecasted sales data: Minimum sum = 36.74533
method represented with ontology fits the sales data to obtain the product life cycle curve, and computes both years to obsolescence and life cycle stages based on the life cycle curve [6]. Compared to previous methods such as ordinal scale based approaches [4,5] that rely on unquantifiable technological attributes, the primary feature of the method is that it can capture market trends for making obsolescence forecasting. In addition, the method includes follow on steps to realize more accurate obsolescence forecasting, including data mining based approach [8] that determines the time range of the zone of obsolescence using data mining of historical data and a leading indicator method [9] to further identify a leading indicator product among particular types of product with a certain life cycle pattern. These methods have some increase in accuracy of forecasting, but at the expense of increased complexity. The method adopted by this paper has been demonstrated as a simple but effective method in the application of forecasting obsolescence of electronic parts [6]. Product life cycle ontology is defined. It contains important life cycle stage information, and can be reused in obsolescence management. The formalized obsolescence forecasting method by ontology representation has potential for integrating with other noncommercial and commercial obsolescence management tools through unified information format.

Ontology representation for obsolescence forecasting is a promising approach to knowledge representation for obsolescence management. Today’s obsolescence management tools are populated by a set of data and service providers that are neither integrated with each other, complete, consistent in the information provided, or even contain consistent definitions of the quantities tracked and archived. It is unclear if the data available is a complete description of the state of a part and data conflicts are more common than data agreement. The ultimate success of the obsolescence management will rely on developing a more integrated web-centric system that allows information sharing, reuse, and collaboration on obsolescence issues across different organizations. The challenge for such an integrated web-based environment is to represent domain knowledge in a manner that supports sharing information from the obsolescence domain integration of all obsolescence resolution activities. Hence, our future work will extend the ontology beyond the initial representation for obsolescence forecasting, to a more comprehensive knowledge representation scheme and management system that can facilitate information sharing and collaboration for obsolescence management and mitigation efforts between existing tools and across different organizations.

Acknowledgment

This work was funded by the National Science Foundation through Grant Nos. 0928530, 0928628, and 0928837. Any opinions, findings, and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References


