Forty Sixth CIRP Conference on Manufacturing Systems 2013

Disassembly liaison graphs inspired by word clouds

Robert J. Riggsa, S. Jack Huab,∗

aIndustrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA
bMechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA
∗Corresponding author. Tel.: +1-734-615-4315; fax: +1-734-647-7303. E-mail address: jackhu@umich.edu.

Abstract

Liaison or connection graphs depict physical mates between components in a graphical representation but do not incorporate any precedence relations or order of assembly or disassembly of components. For the context of disassembly, we developed a method to graphically show not only the physical mates between components but also the disassembly precedence relations amongst all the components. The transformation of a liaison graph into a weighted liaison graph (WLG) is inspired by the generation of word clouds from the visual design domain where component nodes are weighted and colored to depict disassembly precedence relations. A WLG allows users to quickly comprehend the order of disassembly and component embeddedness.

© 2013 The Authors. Published by Elsevier B.V.
Selection and peer-review under responsibility of Professor Pedro Filipe do Carmo Cunha

Keywords: Disassembly; liaison graphs; word clouds

1. Introduction

There is increasing global consciousness and practice for responsible end-of-life (EOL) treatment of used products in a sustainable way. Some accepted methods for responsible EOL treatment are recycling raw materials and remanufacturing or direct reuse of existing components or assemblies. Many companies have created businesses around remanufactured or refurbished products either due to requirement by law, such as the case in the European Union, or because they are able to extract enough economic value out of products that have reached their EOL. Products that are to be disassembled and remanufactured into “like new” products typically undergo grading, disassembly, and cleaning steps before they are to be reassembled into new products. These additional steps can be economically burdensome because they are typically done manually and many components have little to no residual value if not properly disassembled. Due to the economic constraints of the disassembly process, many researchers have proposed methods to determine the optimal level of disassembly either for a single or multi objective criterion [1, 2, 3].

Some components or modules that have high economic value can be embedded into the assembled product and multiple disassembly steps may be required to remove that component or module from the assembly. Additionally, this component or module with high economic reuse value may be physically connected or mated to multiple other components and its removal from the assembly can impact these components. A disassembly precedence graph, a directed graph that shows the order of disassembly for the product, and a liaison graph, a graph that shows the connection or liaisons between components or modules in the assembly, can each show some of the required information for disassembly order and the relation components have with one another by connection or mating points but not in the same graph. In this paper we propose a method to transform a disassembly liaison graph into a weighted liaison graph (WLG) by weighting nodes based on information from the disassembly precedence graph for relative order of disassembly. Our method is inspired by the generation of word or tag clouds from the visual design domain.

Graph theory in the field of mathematics is very simply the study of graphs, which are used to show pairwise relations between features in the graph. For
liaison graphs, circles represent components in the assembly and the edges represent physical connections between components. In a precedence graph, the relationship shown is not what components are physically connected but the precedence order between components. De Fazio and Whitney [4] were some of the first researches to use a liaison graph to show the graphical representation of all valid liaison sequences. They used the idea of liaisons being created between two components to express the assembly sequence order. Kuo et al. [5] utilized liaison graphs, or component-fastener graphs as referred to in the paper, for the purpose of splitting the graph into subgraphs for the end result of creating a disassembly tree. Many researches utilize liaison graphs as an input to their methods or models to show component to component relationships. 

There are related works associated with transforming liaison graphs for different objectives. Sukhan Lee [6] created an abstract liaison graph for the purpose of sub-assembly identification to aid in automatic generation of assembly sequences. Dong et al. [7] developed a method to create a hierarchical attributed liaison graph for the purpose of automatic generation of disassembly sequences. Lee et al. [8] utilized the hierarchical structure method developed by Dong et al. as an input to their model to find the optimal remanufacturing strategy to satisfy environmental regulations and maximize profit. The method we developed seeks to show both the physical connections of components/modules in an assembly and the order of disassembly for each component/module in a single visual layout.

The remainder of the paper is organized as follows: section 2 introduces the concept of word clouds and the method we used to generate a weighted liaison graph, section 3 presents an example using the disassembly of a laptop computer from the literature, section 4 contains discussion and future work, and section 5 presents the conclusions.

2. Word clouds and weighted liaison graph generation

The purpose of word clouds is to visually represent text data in a visually coded way to quickly show the person viewing the information the relative importance of each phrase or word from a much larger text input [9, 10]. Words or terms that are most prominent in a text are represented with larger font and sometimes highlighted with a different font color. Word clouds in the keyword metadata domain can display visually the importance or weight of words or phrases. Figure 1 contains an example word cloud taken from the Center for the Integration of Research, Teaching and Learning (CIRTL) website. The purpose of the word cloud in Fig. 1 is to quickly reveal to the person viewing it what the CIRTL organization is mainly about and their core ideas.

Noah Iliinsky and Julie Steele, the authors of the book Designing Data Visualizations [11], argues that presenting information visually is important and useful because people are “extremely well built for visual analysis,” they can quickly comprehend visual information if presented correctly, and visualizations gives them “actionable insights.” These concepts can also be applied to the visual design of disassembly liaison graphs.

The technique of word cloud generation for text based information can be used for the generation of disassembly liaison graphs to represent various
information pertaining to disassembly decision making. Liaison graphs, or connection graphs, only show the physical connections or mates between components/modules of an assembly. Liaison graphs do not show how embedded components are in the assembly or how many components/modules or tasks that need to be completed before a particular component/module can be removed from the core. The input required for the transformation of a disassembly liaison graph is a disassembly precedence graph. Many methods exist for creation of precedence graphs and finding feasible sequences for either assembly and disassembly and the direction of precedence graph generation has moved towards a highly computer aided approach [12]. Figure 2 contains an example of a liaison graph and Fig. 3 displays the disassembly precedence relations that correspond to the liaison graph in Fig. 2. A disassembly precedence graph shows the general order of disassembly of the components that make up the assembly. The precedence graph in Fig. 3 will be used to illustrate the process of breaking a disassembly precedence graph down into “levels” so as to eventually transform a liaison graph.

Fig. 2. Example liaison graph

Fig. 3. Example disassembly precedence graph

The disassembly precedence graph in Fig. 3 is constructed in such a way that the nodes represent components/modules that need to be removed from the core and each node is associated with a disassembly task time. A disassembly precedence graph can break down all disassembly tasks into individual task steps but we represent each disassembly task as all the task steps required to remove a component/module as a single node. The basic understanding of how to interpret this disassembly precedence graph is that the removal of component A must be done before the removal of components B and C. Component D can be removed only after B is removed from the core, but E depends on the removal of both components C and B, and so forth. It is obvious when looking at Fig. 3 that there are 4 tiers or levels to the disassembly precedence graph. Figure 4 illustrates the breakdown of these 4 levels.

Fig. 4. Disassembly precedence graph broken down into levels

The levels shown in the disassembly precedence graph will be used to transform the liaison graph where earlier levels will be weighted larger and later levels weighted smaller to show precedence relationships in the liaison graph. The rules for level creation are as follows:

- Nodes in the same level will have at least one predecessor that is common, e.g., B and C both have A as their predecessor.
- Nodes in the same level will have at least one common immediate successor node or an eventual successor node, e.g., B and C share a common immediate successor node of E while D and E have no immediate or eventual successor node in common.
- Transient node sets will be given an index and shown in a different color. Transient node sets are nodes in succession that have no common successor nodes with other transient node sets. There are two transient node sets from the previous example, D and F is a transient node set and E and G is the other transient node set.

Based on the rules listed, Fig. 4 can be altered to reflect the established rules for levels and transient node sets and this is shown in Fig. 5.

Fig. 5. Disassembly precedence graph broken down into levels
Nodes D and E are in the same level, level 3, but have a different identifier to show they are in different transient node sets, a and b. The disassembly of components D and F cannot be completed until component B is removed from the core, and the same goes for the disassembly of E and G, except nodes D and F have no impact on the disassembly of nodes E and G, which makes them transient node sets. The information gained from breaking the disassembly precedence down into levels and transient node sets can be fed into an equation to transform the liaison graph in Fig. 3. The equations are the following:

\[ x_i = u \]  
\[ x_y = l \]  
\[ d = \frac{u - l}{y - 1} \]  
\[ x_i = u - d(i - 1) \quad \forall \ i \neq 1, y \]

where

- \( u \): upper bound for largest node size
- \( l \): lower bound for smallest node size
- \( y \): total number of levels in the disassembly precedence graph
- \( d \): node size increment
- \( x_i \): node size for each level
- \( i \): index for each level

Equation (1) will be true for all nodes in level 1 in that they will have the largest allowable node size. Equation (2) will be true for all nodes in the final level, level \( y \), in that they will have the smallest allowable node size. Equation (3) calculates the increment size for the nodes with levels in between 1 and \( y \). Equation (4) computes the node size for all levels \( i \) not equal to 1 or \( y \). The information for transient node sets and the colors used to identify them will be directly passed to the liaison graph. Figure 6 shows the transformed WLG.

3. Example: disassembly of a laptop

The laptop example is taken from a CIRP keynote paper by Hu et al. [13] and it is originally presented in the context of mixed-model assembly but this example is altered to reflect the scenario of disassembling a laptop computer. Figure 7(a) shows an exploded view of all the laptop components and Fig. 7(b) displays the liaison graph. There are 13 total components in the laptop example, A-M. Figure 7(c) shows the disassembly precedence graph for the laptop computer.
Based on the method described in section 2, the levels and transient sets can be identified and fed into the series of equations to determine how the nodes should be weighted and colored in the liaison graph. The levels and transient sets are shown in Fig. 8 and the transformed liaison graph shown in Fig. 9.

4. Discussion and future work

It can immediately be seen from the liaison graph in Fig. 9 that node G, the battery, is the largest node and should be removed first, followed by nodes A and D. It is also clear that nodes A, B, and C belong to a common transient node set and that A must be removed before B and C can be removed. Node H is the smallest node, therefore component H will be removed after all other components/modules have been removed in its common node set. Some improvements to this method can be focused on how to best show immediate successor nodes that do not necessarily follow the rule that all nodes of larger size must be removed from the core before it can be removed. For example, based on Fig. 9 node K is smaller than nodes G, D, F, I, J, L, and M but the task to remove component K only has to follow the removal of G, D, and L. Based on the disassembly precedence graph in Fig. 8, K is a direct successor of L but not the other nodes in the same level as L; therefore, K can be removed from the core before the removal of the other nodes in the same level as L. To show this relationship, a directional liaison can be utilized between nodes L and K to show K is only dependent on the removal of component L and not the other components in the same level as L.

Additional future work can focus on the relationship transient nodes have with other individual nodes to utilize the creation of sub-assemblies for disassembly. For example, nodes A, B, C, and D could remain assembled as a monitor-keyboard sub-assembly for some products at their EOL because the components together meet some criteria for direct reuse into remanufactured products. Because nodes A, B, and C are a transient node set and only have liaisons that connect them to node D, they can be grouped into a sub-assembly. Exploration of how to best find and visually show the relationship transient node sets have with other nodes to can be a useful tool for disassembly system designers.
An example of such a visual method that is inspired by the work of Gupta et al. [14] from a product family-based approach for assembly is shown in Fig. 10.

![Laptop liaison graph with option based sub-assembly identification](image)

With the visual representation in Fig. 10, the dashed liaison lines represent an option based liaison connection where a product can either be fully disassembled into all single components or the sub-assembly consisting of A, B, C, and D can be left together as a single sub-assembly for reuse. Since node D has a node size greater than or equal to nodes A, B, and C and is a part of the same set as all other nodes, the disassembly level for the sub-assembly A, B, C, and D will be in the same level as D.

A limitation of a WLG is how many components/modules can be weighted and shown in a single graph. The laptop example has 13 components and there are only 6 different node sizes shown in the graph so the difference in node size is easily determinable. The point to which human beings can no longer easily decipher node size to determine disassembly order in a WLG is not known but should be investigated. Since there is this limitation, future work can also focus on methods to present a WLG for the case where there are too many levels to show in a single WLG.

5. Conclusion

This paper presents a method to visualize the disassembly sequence information from a liaison graph and disassembly precedence graph into a single weighted liaison graph. Previous approaches to augmenting liaison graphs do not consider precedence information and our approach is the first to combine the information from both into a single graph. A WLG allows users to quickly and efficiently determine what components/modules are in physical contact with one another and their relative disassembly order with respect to one another. The method presented in this paper is inspired by the creation of word clouds from the visual design domain and it is the authors’ hope that this method can aid disassembly decision makers in both visualizing for themselves and showing others key characteristics for disassembling products.

Acknowledgements

The authors would like to thank the National Science Foundation (NSF) for supporting Robert Riggs with the Graduate Research Fellowship Program (GRFP).

References

[1] Duflou JR, Seliger G, Kara S, Umeda Y, Ometto A, Willems B (2006) Efficiency and feasibility of product disassembly: A case-based study. CIRP Annals 57(2):583-600.
[2] Kara S, Poropratap P, Kaebennick H (2006) Selective Disassembly Sequencing: A Methodology for the Disassembly of End-of-Life Products. CIRP Annals 55(1):37-40.
[3] Giudice F, Fargione G (2007) Disassembly planning of mechanical systems for service and recovery: a genetic algorithms based approach. Journal of Intelligent Manufacturing 18(3):313-329.
[4] De Fazio TL, Whitney DE (1987) Simplified Generation of All Mechanical Assembly Sequences. IEEE Journal of Robotics and Automation 3(6):640-658.
[5] Kao TC, Zhang HC, SH (2000) Disassembly Analysis for electromechanical products: a graph-based heuristic approach. International Journal of Production Research 38(5):993-1007.
[6] Lee S (1994) Subassembly Identification and Evaluation for Assembly Planning. IEEE Transactions on Systems, Manufacturing, and Cybernetics 24(3):493-503.
[7] Dong T, Zhang L, Tong R, Dong J (2006) A hierarchical approach to disassembly sequence planning for mechanical product. International Journal of Advanced Manufacturing Technology 30(5-6):507-520.
[8] Lee HB, Cho NW, Hong YS (2010) A hierarchical end-of-life decision model for determining the economic levels of remanufacturing and disassembly under environmental regulations. Journal of Cleaner Production 18(13):1276-1283.
[9] Lehmenn S, Ziegler J, Tetzlaff L (2009) Comparison of tag cloud layouts: task-related performance and visual exploration. Proceedings of IFIP TC 13 International Conference 392-404.
[10] Viegas FB, Wattenber M (2008) Tag Clouds and the Case for Vernacular Visualization. Interactions 15(4):49-52.
[11] Ilin斯基 N, Steele J. Designing Data Visualizations. O’Reilly Media: 2011.
[12] Kanai S, Takahashi H, Makino H (1996) ASPEN: Computer-aided assembly sequence planning and evaluation system based on predetermined time standard. CIRP Annals 45(1):35-39.
[13] Hu SJ, Ko J, Weyland L, ElMaraghy HA, Lien TK, Koren Y, Bley H, Chryssolouris G, Naar N, Shpitalni M (2011) Assembly system design and operations for product variety. CIRP Annals 60(2):715-733.
[14] Gupta S, Krishnan V (1998) Product family-based assembly sequence design methodology. IIE Transactions 30(10):933-945.