SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 5 No. 6
ISSN 2054-7390

Disassembly Modeling of an of End-Of-Life (EOL) Mechanical Damper for Recycling

[1]Akintunde, Olalekan. Akintunde, [2]Simolowo, Oluwafunbi. Emmanuel
[1, 2] Department of Mechanical Engineering, University of Ibadan. Nigeria.
esimmar@yahoo.com; oe.simolowo@mail.ui.edu.ng

ABSTRACT

Today's rapidly developing technologies and product designs have enabled manufacturers to deliver new products to consumers at a dramatic rate. This has in turn resulted in shorter lifespan for products, because, more often than not, they are discarded even though they are still in excellent working conditions. An overflowing stream of used scrapped products has become an alarming problem for waste management, as it is quickly elevating the level of environmental detriments. Countries around the world have also observed an explosive growth in the waste stream that is filling up municipal landfills and clogging up incinerators. These problems, in addition to the limited material resources on Earth have made the recycling of End-Of-Life (EOL) products a rapidly expanding research area. However, before recycling of EOL products can be done, they need to be disassembled. Therefore, this paper focuses on the automation of disassembly of EOL car dampers that are found wasting away in many auto repair workshops. It discusses the application of the component-mating graph and newer CAD methods in the disassembly modeling of an EOL mechanical damper.

Keywords: Disassembly, Component-Mating Graph, Modularity Analysis, Sequencing

1 Introduction

In the recent years, the knowledge of conserving energy, material resources, and landfill (dump sites) capacity, and recycling regulations, has put pressure on many manufacturers to produce and dispose of products, in an environmentally friendly manner. It has also aroused consumer interest in "green products", and has promoted environmentally responsible use, consumption, and disposal of products. Many governments worldwide are also stiffening legislation, thereby requiring manufacturers to use as much recycled material as possible, and to play a key role in recycling products at their end of life (EOL) [1] – [6].

Throughout the world, many reclamation facilities have been established by product manufacturers, for study and disassembly of their products. Sony Corporation (a Japanese Electronics Manufacturer), has built the Sony Disassembly Evaluation Workshop, at Stuttgart, Germany, to assess the reuse and recycling qualities of their electronic products. The International Business Machines (IBM) Corporation, which is one of the world's largest manufacturers of computers, has also established the Reutilization Centre at Endicott, New York, to disassemble and recover reusable components from their personal and notebook computer products. Even though a significant amount of research has focused on the design of products

DOI: 10.14738/tmlai.56.2530
Publication Date: 1ST Oct, 2017
URL: http://dx.doi.org/10.14738/tmlai.56.2530

from an environmental and disassembly perspective [9] – [10], disassembly still plays an important role in material and product recovery. [11].

1.1 Objective of Study

The objective of this work is to apply the component-mating graph and modern CAD methods to a newly-analysed EOL mechanical device, a car damper that is commonly disposed. This entails the use of: Autodesk Inventor, Computer Aided Design (CAD) software; MATLAB; SimMechanics Link Utility software for linking the CAD software with MATLAB for disassembly modeling; graph representation of the car damper and modularity analysis.

2 Methodology

One of the modeling methods is the Component-Mating Graph, through which the Modularity Analysis of the Damper is performed, in this paper. This is also referred to as Component-Fastener Graph. It is an undirected graph which can be constructed using data from Computer Aided Design (CAD) software that was used to design the product. For this paper, Autodesk Inventor Professional 2013 software was used. In a Component Mating Graph [12] – [13], $G = (V,\ E)$, where vertices $V = \{v_1,\ v_2,\ \ldots,\ v_n\}$ represent components; and edges$_{\text{n}}$ denote geometrical relationships among components, where m is the number of edges. It is clear that the upper limit for $|V|$ is n. Figure 1 shows the CAD model of the Damper, with its four component parts. The labels (shown) in parenthesis are chosen arbitrarily to make the components identifiable during the disassembly modeling process. The components of the Damper model were mated/ constrained, and their motion was simulated within Autodesk Inventor. An XML file was generated from the CAD model within Autodesk Inventor, and this new file was exported to, and saved in MATLAB's current directory

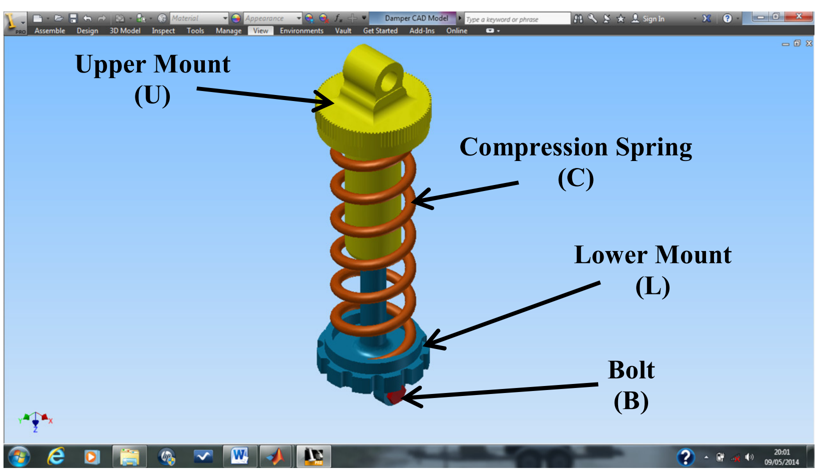

Figure 1: Damper CAD Model

At the command prompt in MATLAB, the **mech_import** command was executed. This brought up the **Import Physical Modelling XML** dialog box, and the previous XML file was selected. A SimMechanics model of the damper was then generated in MATLAB. This model showed the connection between the parts of the imported 3D CAD model, as well as their various properties, such as; mass, principal moments of inertia, volume and surface area. The SimMechanics model is shown in Figure 2.

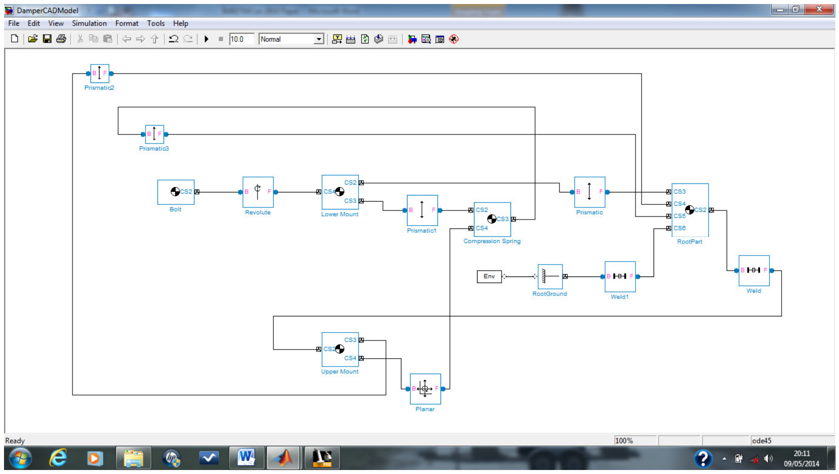

Figure 2: SimMechanics Model of the Damper

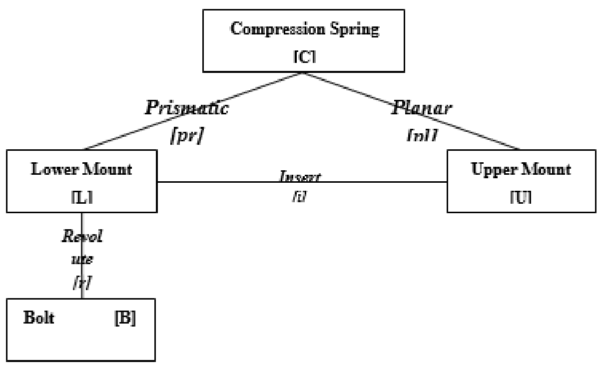

Figure 3: Component-Mating Graph of the Damper

3 Application of Component-Mating Graph to Damper

The Connection Graph (shown in Figure 3) was then constructed from the SimMechanics model by excluding the additional feature blocks created by the software The Connection Graph of Figure 3 can be written mathematically as: $G_C = (V,\ E)$,

Where: $V = \{U,\ C,\ L,\ B\}$, i.e. (set of vertices), $E = \{pr,\ pl,\ i,\ r\}$, i.e. (set of edges)

Number of vertices, or components $(n) = 4$. Let $E_C = [E_{ij}]$ be $G_C's$ adjacency matrix.

$$\therefore E_C = \begin{array}{c c} & \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \text{ Were } E_{ij} = \begin{cases} 1, if\ component\ i\ is\ connected\ to\ component\ j \\ \qquad 0,\ otherwise \end{cases}$$

3.1 Modularity Analysis

To reduce the complexity of an EOL product, an assembly (module) can be decomposed into several subassemblies (sub-modules). The subassemblies can be further decomposed into simple subassemblies. Before describing the decomposition methodology, the concepts of cut-vertices, pendant vertices, and bi-connected graphs must be defined. Cut-Vertex (CV): a vertex whose removal disconnects the graph. In a module, cut- vertices refer to the connection component between two other components. Pendant Vertex: only one edge incident on the vertex is a pendant vertex. In a module, it represents a single component. Bi-connected graph: a connected graph with no cut-vertices. In a module, a bi-connected graph means a sub-module.

3.2 Subassembly

There are three types of geometric assembly methods: Type I: An assembly which has no main component (or CV). In this type of assembly method, all the components are assembled with others. Type II: An assembly which has a main component (or CV). In this type of assembly method, other components or subassemblies are directly assembled or indirectly assembled with the main component. Type III: A combination of Type I and Type II. From the Component Mating Graph of Figure 3, it is obvious that the Damper contains the Type II subassembly category. When considering the disassembly processes, the following disassembly rules are adopted: Rule 1: If a type I subassembly is found, the subassembly can be further disassembled as sub subassemblies, or single components. Rule 2: If a type II subassembly is found, the subassembly can only be further disassembled as single components. Rule 3: If a type III subassembly is found, the subassembly can be further disassembled into more subassemblies and/ or single components.

3.3 Depth First Search (dfs)

The dfs algorithm used in this paper was adopted from the algorithm shown in the book *of Algorithmic Theory and Practice* (Gilles and Paul, 1988). The search was done by following a path from the top to the bottom component of the Damper.

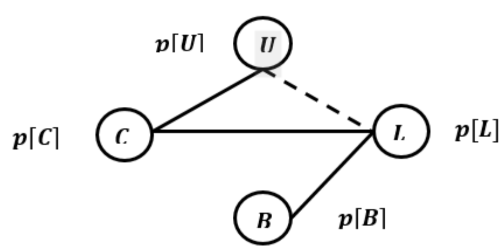

Figure 4: Depth First Tree (G_d) for the Component Mating Graph (G_C) of the Damper

The label: $prenum\ [v_i]$, which appears on each node is the algorithm for the dfs procedure, and is illustrated as follows:

$prenum\ [1]$	$dfs(U)$	Initial call
$prenum\ [2]$	$dfs(C)$	Recursive call
$prenum\ [3]$	$dfs(L)$	Recursive call

$prenum\ [4]$ $dfs(B)$ Recursive call

The broken line represents an edge that was present in graph G_C, but which was not present in graph G_d.

3.4 Cut Vertex Search

From the tree G_d of Figure 4, the $lowest[v_i]$ for each node is calculated in post order (i.e. from last node to first node of the dfs procedure). This is given by the relation:

$$lowest[v_i]\ or\ L[v_i]\ = min\left\{prenum[v_i],\ \ min\{prenum[w_i]\},\ \ min\{prenum[c_{ij}]\}\right\}.$$

Where: $prenum[v_i]$: the search order for each node v_i (from dfs procedure)

$prenum[w_i]$: if a node v_i has a back edge (shown with dashed line) to one or more node(s) (say v_k) higher up the tree in G_d , then $prenum[w_i] = min\{prenum[v_k]\}$ for the node(s) adjacent to only the back edge; otherwise, $prenum[w_i] = \infty$.

$prenum[c_{ij}]$: for each child j of node v_i in G_d, $prenum[c_{ij}] = lowest[v_j]$ for all the children. If node v_i has no child, then $prenum[c_{ij}] = \infty$.

The results are shown below:

$$lowest[B] = min\left\{prenum[B],\ \ min\{prenum[w_B]\},\ \ min\{prenum[c_{Bj}]\}\right\}$$
$$= min\{4, \infty, \infty\}\ \ = 4.$$
$$lowest[L] = min\left\{prenum[L],\ \ min\{prenum[w_L]\},\ \ min\{prenum[c_{Lj}]\}\right\}$$
$$= min\{3, 1, 4\}\ \ = 1.$$
$$lowest[C] = min\left\{prenum[C],\ \ min\{prenum[w_C]\},\ \ min\{prenum[c_{Cj}]\}\right\}$$
$$= min\{2, \infty, 3\}\ \ = 2.$$
$$lowest[U] = min\left\{prenum[U],\ \ min\{prenum[w_U]\},\ \ min\{prenum[c_{Uj}]\}\right\}$$
$$= min\{1, 3, 2\}\ \ = 1.$$

Figure 5 shows the comparison between $prenum[v_i]$ and $lowest[v_i]$ for all the nodes of G_d.

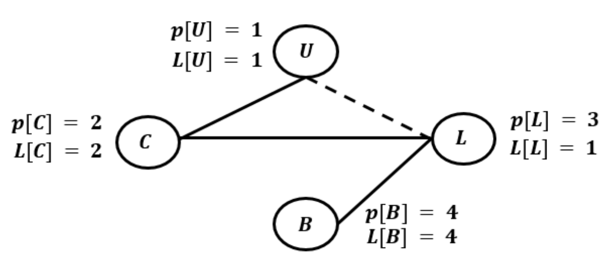

Figure 5: Comparison of $prenum[v_i]$ and $lowest[v_i]$ for Nodes of Graph G_d of the Damper

The cut vertex is determined from the following rules:

Rule 1: The root (first vertex) of graph G_d is a cut vertex of G_C if and only if it has more than one child. Since the root vertex $[U]$ has only one child $[C]$, it is not a cut vertex.

Rule 2: A vertex v_i other than the root of G_d is a cut vertex of G_C if and only if v_i has a child j such that $lowest[v_j] \geq prenum[v_i]$. From Figure 4.8, the only vertex that satisfies this condition is $[L]$.

Therefore, the Cut Vertex (CV) of the graph G_d is vertex $[L]$.

3.5 Pendant Vertex and Sub-Graph Classification

To find the pendant vertex, it can be assumed that $CV's$ row and column vectors are zero in the adjacency matrix E_{ij}. If the total edge number of a vertex (except for the edge that is connected with CV) is 0, then that vertex is a pendant vertex based on the CV.

Recall:

$$\therefore E_C = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

For any vertex i, if $EN_i = \sum_{j=1}^{n} E_{ij} = 0$, where $j \neq CV$, then i is a pendant vertex.

From the matrix E_C above, only vertex ID 4 satisfies this condition.

$$EN_4 = \sum_{j=1}^{4} E_{4j} = E_{41} + E_{42} + E_{43} + E_{44} = 0 + 0 + 0 + 0 \quad = 0.$$

Therefore, the Pendant (p) Vertex is vertex $[B]$. To find the sub-graphs in the graph G_C, it can be assumed the $CV's$ row and column vectors are zero in the adjacency matrix E_{ij}. This means that the CV and p vertices are completely removed from the graph G_C. To achieve this, the dfs procedure will be carried out on the new graph $G'_C = G_c - CV - p$, as shown in Figure 6. The algorithm is as shown:

$prenum\ [1]$	$dfs(U)$	initial call
$prenum\ [2]$	$dfs(C)$	recursive call

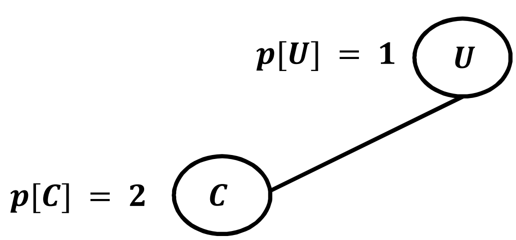

Figure 6: Depth First Tree $(G'_C = G_c - CV - p)$ of the Car Damper

From the dfs procedure of Figure 6, there is one sub-graph (subassembly) within the graph G'_C, and this is denoted by: $S = \{U, \ C\}$.

3.6 Decomposition and Modularity Analysis

Introducing CV and p back into the graph G'_C, and doing some re-arranging, produces the modularity graph shown in Figure 7.

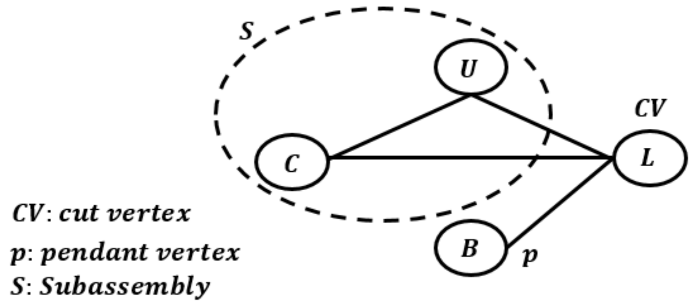

Figure 7: Modularity Analysis I of the Damper.

The next step is to determine whether $CV \subset S$. When examining the total number of edges for the subassembly (S) that are connected with CV, the CV will be grouped with the subassembly that has the highest number of edges connected with it. From Figure 7, subassembly S has two edges connected with the CV. Therefore $S = \{U, \ C, \ L\}$, as shown in Figure 8. Since the Damper module has now been decomposed into sub-modules, the disassembly process is analyzed by the DPMs. Disassembly precedence means a component i cannot be removed until component j is removed. The precedence relation is local to the parts concerned, and signifies the partial order of disassembly. In this paper, six disassembly directions are adopted (Eq. (1) – (6)): $\pm x$, $\pm y$, and $\pm z$ (from the CAD model of Figure 1).

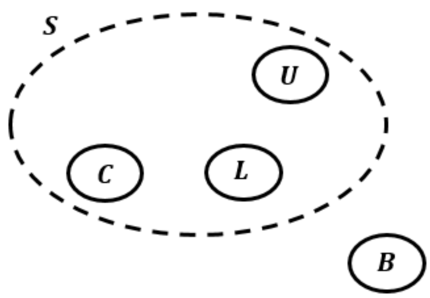

Figure 8: Modularity Analysis II of the Damper.

3.7 Disassembly Precedence Matrices (DPMs) Analysis

$$DP_{\pm d} = [DP_{ij}] \begin{cases} 1, component\ j\ needs\ to\ be\ removed\ before\ component\ i, along\ \pm d \\ 0, otherwise \end{cases}$$

(1), (2)

$$DP_{+x} = \begin{array}{cc} & \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad DP_{-x} = \begin{array}{cc} & \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \qquad (3), (4)$$

$$DP_{+y} = \begin{array}{cc} & \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad DP_{-y} = \begin{array}{cc} & \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (5), (6)$$

$$DP_{+z} = \begin{array}{cc} & \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad DP_{-z} = \begin{array}{cc} & \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

3.8 Merging of the DPM's

The six DPM's can be written as (Eq. (7) – (12)):

$$DP_{+x} = \begin{array}{ccc} & \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} & \boldsymbol{D_{+x}} \\ \begin{array}{c} \mathbf{S} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 0 \end{array} \end{array} \qquad DP_{-x} = \begin{array}{ccc} & \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} & \boldsymbol{D_{-x}} \\ \begin{array}{c} \mathbf{S} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \end{array} \end{array} \qquad (7), (8)$$

$$DP_{+y} = \begin{array}{ccc} & \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} & \boldsymbol{D_{+y}} \\ \begin{array}{c} \mathbf{S} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 1 \end{array} \end{array} \qquad DP_{-y} = \begin{array}{ccc} & \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} & \boldsymbol{D_{-y}} \\ \begin{array}{c} \mathbf{S} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 1 \end{array} \end{array} \qquad (9), (10)$$

$$DP_{+z} = \begin{array}{ccc} & \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} & \boldsymbol{D_{+z}} \\ \begin{array}{c} \mathbf{S} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 1 \end{array} \end{array} \qquad DP_{-z} = \begin{array}{ccc} & \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} & \boldsymbol{D_{-z}} \\ \begin{array}{c} \mathbf{S} \\ \mathbf{B} \end{array} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 1 \end{array} \end{array} \qquad (11), (12)$$

Therefore, the six DPM's can be represented as one DPM D_{ij} as shown in Eq. (13)

$$DP = \begin{array}{cc} & \begin{array}{cccccc} \boldsymbol{D_{+x}} & \boldsymbol{D_{+y}} & \boldsymbol{D_{+z}} & \boldsymbol{D_{-x}} & \boldsymbol{D_{-y}} & \boldsymbol{D_{-z}} \end{array} \\ \begin{array}{c} \mathbf{S} \\ \mathbf{B} \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \qquad (13)$$

From the matrix "DP", the subassembly or component can be disassembled if $DP_{ij} = 0$. Therefore, subassembly $S = \{U,\ C,\ L\}$ can only be disassembled from the $-x$ direction, and component $\{B\}$ can only be disassembled from the $+x$ direction.

4 Result of Disassembly Analyses

The disassembly tree and sequence are obtained as a final outcome of the disassembly analyses of the car damper as shown in Fig. 9. From the disassembly tree, in level 0, the Damper assembly is represented as a parent vertex $\{U, \ C, \ L, \ B\}$ which is in the level 0 (L_0). In level 1 (L_1), the assembly is decomposed into one subassembly $S = \{U, \ C, \ L\}$, and component $[B]$. The Modularity Analysis and Disassembly Precedence Matrix (DPM) test are processed recursively until all the components are disassembled from the whole assembly. In level 2, the subassembly S is decomposed into components $\{U\}, \ \{C\}, \text{ and } \{L\}$. Thus, the disassembly process is complete.

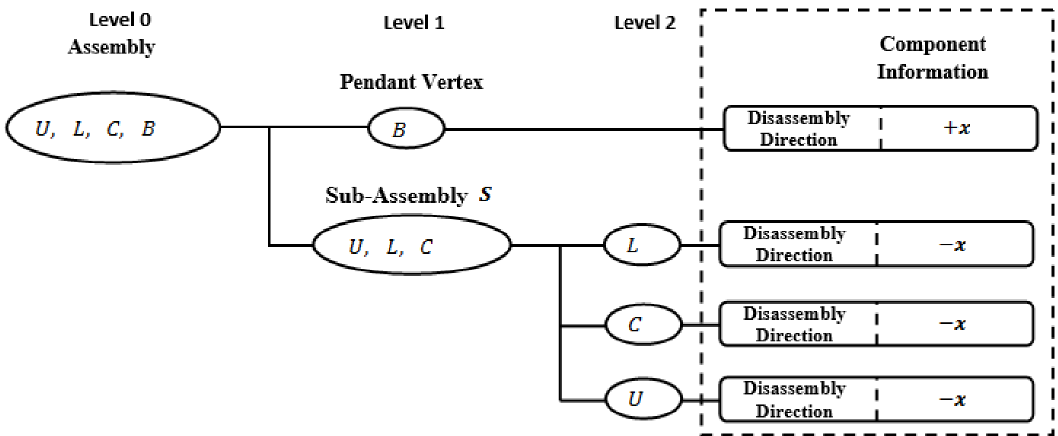

Figure 9: Disassembly Tree Representation of the Car Damper

5 Conclusion

A graph-based product design representation is presented for generating disassembly sequences of an Automobile Damper assembly. A Modularity Analysis was also used to generate feasible disassembly sequences. The process involved the use of the Component-Fastener Graph and Disassembly Precedence Matrix (DPM) that were developed for the Damper. The Component-Fastener Graph represents the hierarchy of the product structure, while the DPM represents the local and partial order of disassembly. The final result of the analysis was the disassembly tree, which shows complete disassembly sequences of the Damper. This sequence of disassembly is to incorporated in the automation of the recycling process of the damper.

REFERENCES

[1]. Chynoweth E. and Rotman D, *Catalyst: Cover story*, June 1993. chemical week,

[2]. Williams, J. A. S., *A Review of Research towards Computer Integrated Demanufacturing for Materials Recovery.* International Journal of Computer Integrated Manufacturing, 2007. 20(8): p. 773-780.

Simolowo, O. E.; Mousavi, A.; Adjapong, P. O., *A Computer-Based Product Classification and Component Detection for Demanufacturing Processes*. International Journal of Computer Intergrated Manufacturing Tailor and Francis, 2011. 24(10): p. 900 – 914.

[3]. Simolowo O. E. and Mousavi A., *Developing Imaging Software for Automated Recycling Processes (Phase 1).* Journal of Advanced Materials Research, 2012. 628: p 298 -303.

[4]. Simolowo O. E. and Onovugbe E. C., *Automation of Generation of Models for Disassembly Process Planning for Recycling* Book Of Proceeding Conference of Mechanical Engineering ICME_2013 at Imperial College London, 3, p. 1964 – 1968.

[5]. Onovughe, E. C. I., *Disassembly Modelling Process for Recycling of EOL Products*". Unpublished. Project, University of Ibadan, Ibadan, Nigeria, 2012 p. 9-31.

[6]. Li, W.; Zhang, H. P.; Wang, B.; and Awoniyi, S. A., *Design for Disassembly Analysis for Environmentally Conscious.* Design and Manufacturing. Manufacturing Science and Engineering, ASME, 1995; 2-2, 3(2): p. 969-976.

[7]. Capelli et al. *Design for disassembly: a methodology for identifying the optimal disassembly sequence.* Journal of Engineering Design 2007, 18(6): p. 563–575.

[8]. Gu, P. and Yan, X. (1996). *CAD-Directed Automatic Assembly Sequence Planning*. Int. J. PROD. RES., 1996, 33(11): p.3069-3100.

[9]. Mazouz, A. K.; Souilah, A.; and Talbi, M. *Design of an Expert System for Generating Optimal Assembly Sequence*. Computer-Aided Engineering Journal, December, 1991: p. 225-259.

[10]. Tang, Y.; Zhou, M. C.; Zussman, E.; and Caudill, R. (2002). *Disassembly Modeling, Planning, and Application.* Journal of Manufacturing Systems. 2002, 21(3): p. 200-217.

[11]. Gilles Brassard and Paul Bratley. *Algorithmic Theory and Practice.* 1988. Prentice hall publishers, Michigan

[12]. Johnson Baugh, R., *Discrete Mathematics Textbook.* 2005. .6th Edition