A. M. Farid and D. C. McFarlane, "An Approach to the Application of the Design Structure Matrix for Assessing Reconfigurability of Distributed Manufacturing Systems," in Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), (Prague, Czech republic), pp. 1–6, 2006.

An Approach to the Application of the Design Structure Matrix for Assessing Reconfigurability of Distributed Manufacturing Systems

Amro M. Farid
University of Cambridge
Institute for Manufacturing
16 Mill Lane, Cambridge, United Kingdom
af320@eng.cam.ac.uk

Duncan C. McFarlane
University of Cambridge
Institute for Manufacturing
16 Mill Lane, Cambridge, United Kingdom
dcm@eng.cam.ac.uk

Abstract

In recent years, a large number of approaches to developing distributed manufacturing systems has been proposed. One of the principle reasons for these developments has been to enhance the reconfigurability of a manufacturing operation; allowing it to readily adapt to changes over time. However, to date, there has only been a limited assessment of the resulting reconfigurability properties and hence it remains inconclusive as to whether a distributed manufacturing system design approach does in fact improve reconfigurability. Recently, the "Design Structure Matrix" has been proposed as a tool for assessing the modularity of elements in a distributed manufacturing system[4]. However, a clear and systematic approach to its application to distributed manufacturing systems has yet to be developed. This paper outlines such an approach in three phases: 1.)definition of system boundary and functionality 2.)identification of system components and finally 3.)identification of interfaces between components. The use of the Design Structure Matrix is illustrated in assessing a robot assembly cell designed on distributed manufacturing system principles.

1 Introduction

Recent trends in manufacturing are characterized by continually evolving and increasingly competitive marketplaces. The effective implementation of lean manufacturing principles, in many instances, had freed excess capacity, and thus gave consumers greater influence over the quality, quantity and variety of products[13][8]. In order to stay competitive, manufacturing firms have had to respond with high variety products of increasingly short product life cycle[11]. In other words, new products must be introduced to the market in ever shorter time and with increasing frequency so as to continually develop the variety of the offered product range. To address these drivers, the concept of reconfigurable manufacturing systems has been proposed[11].

1.1 Distributed Manufacturing as an Enabling Technology for Reconfigurable Manufacturing Systems

Mass customized and short-life cycle products require that capacity be adjusted flexibly with the addition of new production and material handling resources and/or their tooling. Similarly, new product introduction may require that the manufacturing system be rapidly redesigned in terms of a rearrangement of its component production and material handling resources [11]. Each of these reconfigurations require extensive integration effort. At a low level, the mechanical interfaces between production resources, products and material handlers must be addressed. At a high level, each new production resource with its associated tools, fixtures and end-effectors requires integration into the continuous-real-time, discrete event, scheduling, and planning control layers [13]. To enable this rapid integration, a reconfigurable manufacturing system requires distributed or modular, open architecture controllers[11].

1.2 Manufacturing System Scope

This paper restricts its scope to distributed manufacturing systems (DMS). A DMS is a system that uses a collection of value-adding and material-handling resources to transform raw material into finish product via DMS control system. A DMS control system is a system that controls the planning, scheduling, execution and continuous-time control functionality with decision elements distributed among the DMS's value-adding and material handling resources. A conceptual representation of a distributed manufacturing system is shown in Figure 1[4].

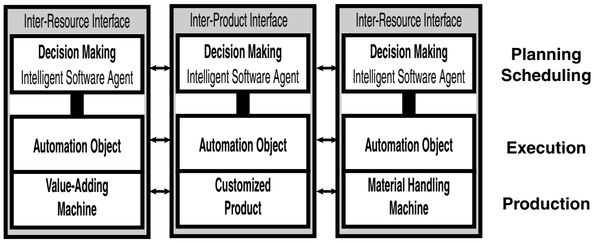

Figure 1. A Conceptual Representation of a Distributed Manufacturing System

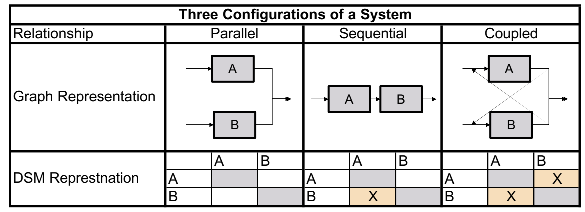

Figure 2. DSM Representations of System Configurations

1.3 The Need for Manufacturing System Architecture Assessment

Assessing the suitability of a DMS to the drivers of mass customization and short product life cycles requires measures of both its behavioral and its structural performance. Measures for the former are well developed in the literature and industry. Among them are throughput, lead time, system flexibility, and responsiveness. Measures of the structural performance, however, have been more elusive. As a result, assessing the reconfigurability of manufacturing systems based upon its structural properties has been only given limited coverage [12].

1.4 Paper Outline

Recently, the "Design Structure Matrix" (DSM) has been proposed as a tool for assessing the modularity of a DMS[4]. This paper details a clear and systematic approach to the application of this tool. In Section 2, the DSM is brief reviewed in the context of modularity assessment. Section 3 forms the body of the paper and explains a three phase approach to the application of the DSM to DMS's. Finally, Section 4 demonstrates the approach on the HCBA robot work cell.

2 Modularity Assessment using the Design Structure Matrix

The design structure matrix is a systems analysis tool that captures the interactions between components of a complex system in a compact and clear representation [3]. Given two components A and B, they may interact in a parallel, serial or coupled fashion. These interactions may be spatial, structural, energy, material or information interfaces [17]. Figure 2 shows the graphical representation of these interactions and their associated design structure matrices. The placement of an off-diagonal "X" represents the existence of an interaction between two components: A and B[3]. Some authors, however, have replaced the "X" with numerical values in order to subjectively assess the strength of a particular interaction [15][5][21]. A tutorial of the DSM can be found at [3] and is formally described in [19].

The DSM has also served as a data structure from which a variety of modularity metrics have been developed. Gershenson has conducted an exhaustive review of these metrics [7] and their associated definitions [6]. Huang and Kusiak have discussed matrix-based modularity metrics to facilitate the realization of highly customized products [10] and this may be directly reapplied to the reconfiguration of manufacturing systems. Matrix-based modularity metrics have also been used to advance the role of modularity in life cycle engineering [9][5][18][22]. Analogously, modularity may play a role in the efficient operation, maintenance, and decommissioning of distributed manufacturing systems.

3 An Approach to DSM Analysis for Distributed Manufacturing Systems

This section explains a three phase approach to the application of the DSM to DMS's: 1.) definition of system boundary and functionality, 2.) identification of system components 3.)identification of interfaces between components.

3.1 Definition of System Boundary & Functionality

DSM's typically analyze closed systems and so the the inherently open DMS must be translated to an analogous closed system. As illustrated in Figure 3, DMS's convert information, energy, and material into other forms of information, waste energy, and material. The DSM is not capable of capturing these interactions with the outside environment which includes the raw material stream. Instead, the products are made as an intrinsic part of the analogous closed system. Additionally, the system boundary acts as an infinite source of the necessary system inputs, an infinite sink to the generated outputs, and serves as a common platform

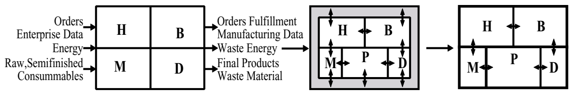

Figure 3. Analogous Open and Closed Distributed Manufacturing Systems

to which all the manufacturing system components can connect. Capturing all of the interactions between the system boundary and the rest of the system adds little value to the analysis. Instead, interactions between two manufacturing system components via the system boundary are treated as direct.

3.2 Identification of System Components

The DMS's functionality can be decomposed into five types of subsystem components:

$L = \{l_1 \ldots l_{s(L)}\}$ – A product line composed of $s(L)$ intelligent products [14]. The s() operator gives the size of a set.

$M = \{m_1 \ldots m_{s(M)}\}$ – A set of $s(M)$ transforming machines.

$H = \{h_1 \ldots h_{s(H)}\}$ – A set of $s(H)$ material handlers.

$D = \{d_1 \ldots d_{s(D)}\}$ – A set of $s(D)$ material handler directors. The primary purpose of material handler coordinators is to prevent the collision of two or more material handlers. Examples include track gates, traffic lights, railroad points etc.

$B = \{b_1 \ldots b_{s(B)}\}$ – A set of $s(B)$ independent buffers. An independent buffer is a manufacturing system artefact that is not physically attached to any transforming machine or transporting material handler and is capable of storing raw material, WIP, or final goods at a specified location.

Each of these subsystems has its own set of associated components.

$C_L = \{\{c_{l1}\} \ldots \{c_{l(s(L))}\}\}$ – Every product l_i has its associated set of components c_{li}.

$C_M = \{\{c_{m1}\} \ldots \{c_{m(s(M))}\}\}$ – Every transforming machine m_i has its associated set of components c_{mi}.

$C_H = \{\{c_{h1}\} \ldots \{c_{h(s(H))}\}\}$ – Every material handler h_i has its associated set of components c_{hi}.

$C_D = \{\{c_{d1}\} \ldots \{c_{d(s(D))}\}\}$ – Every material handling director d_i has its associated set of components c_{di}.

$C_B = \{\{c_{b1}\} \ldots \{c_{b(s(B))}\}\}$ – Every independent buffer b_i has its associated set of components c_{bi}.

The identification of the subsystem components is not trivial and more than one set of component aggregations can be conceived to describe a given subsystem. One approach to identifying the components is to use Axiomatic Design Theory for large flexible systems[20]. Specifically, the functional requirements are a set of transformation, transportation, storage, and material handling coordination processes and they can be allocated flexibly to the transforming machines, material handlers, buffers and material handling directors respectively. These production processes can be further decomposed into sub-functions which must have their corresponding subsystem components. In a complementary approach, Baldwin & Clark identify components based upon the principle of visible design rules[1]; identified subsystem components have clear interfaces between two distinct subsystems. A combination of these two approaches is used here.

*Transforming Machine Components:*A machine must have a tool and a fixture to form and hold the product respectively. Additionally, the machine must have a component for execution control, and an intelligent software component for planning and scheduling activities. Implicitly, the machine must also have a location by which to relate itself spatially to the other manufacturing subsystems. Although the machine location is not strictly speaking a machine component, it, like the other components, can be specified as a set of scalar parameters pertaining to the machine. The set of machine components is then

$$C_M = \{\text{Location, Tool(s), Fixture(s), Execution Controller, Intelligent Software}\}. \quad (1)$$

*Material Handling Components:*Material Handler components can be treated similarly. A material handler must have an end-effector with an associated motion mechanism to move and hold the product. Additionally, the material handler must have a component for execution control, and an intelligent software component for planning and scheduling activities. Implicitly, the machine must also have a region of motion by which to relate itself spatially to the other manufacturing subsystems. The set of material handling components is then

$$C_H = \{\text{Motion Region, End-Effectors(s), Execution Controller, Intelligent Software}\} \quad (2)$$

Material Handling Director Components: A material handler director requires a number of (virtual or physical) “ports” from which material handlers can enter or exit. These ports, if physical, require an execution controller which enables them. Additionally, an intelligent software component is required for the planning and scheduling activities. As with other subsystems, the location of the subsystem is added to the list of components. The set of material handling director components is then

$$C_D = \{\text{Location, Ports(s),} \\ \text{Execution Controller, Intelligent Software}\}. \quad (3)$$

Independent Buffer Components: Independent buffers have a subset of the functionality of machines in that they must store/hold a product but not form it. Assuming that the independent buffer requires active control and has finite capacity, the set of independent buffer components is then

$$C_B = \{\text{Location, Fixture(s),} \\ \text{Execution Controller, Intelligent Software}\}. \quad (4)$$

Intelligent Product Components: The list of product components is simply all the parts of the final product. Also, a number of intelligent software components will be needed to control the planning and scheduling activities of the various subassemblies. The intelligent product components set is

$$C_L = \{\text{Part(s), Intelligent Software(s)}\}. \quad (5)$$

3.3 Identification of Component Interfaces

Having defined the manufacturing system components, the DSM is constructed, as shown in Figure 4 . Capturing

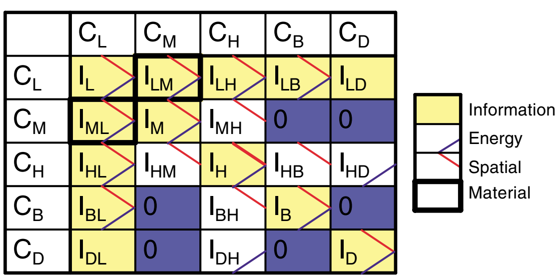

Figure 4. DSM for a DMS Subsystems

all possible component-component interactions is a daunting process even for modestly sized systems. For this reason, it is necessary to identify a priori the matrix elements where there is typically no interaction. In this way, effort can be focused on the nonzero interactions. In distributed manufacturing systems, no $M-B$, $M-D$, $B-D$ interactions seem to exist. Additionally, the interactions in the I_L, I_M, I_B, I_D matrices are said to be entirely intra-subsystem and hence are zero off the block-diagonal, where the block diagonal represents the intra-subsystem interactions. However, I_H must allow spatial interactions between two material handlers. In this case, the safety-conscious manufacturing system designer should install a (physical or virtual) material handling director to control the region of potential collision[16].

The nonzero inter-subsystem interactions should be further classified as spatial, material, information, or energy. In the scope of this paper, structural interactions are not distinguished from energy interactions as they can be viewed as elastic deformation. Information interactions are constrained to within the subsystems and to the products. Energy interactions occur wherever there is possible physical connection or where energy is being transferred as part of transformation processes. Spatial interactions occur wherever there is a possible collision between two subsystems. Material transfer is limited to consumables such as lubricants and coolants and only occur between products and transforming machines.

It is worthwhile to note that this is not the only DSM that can describe this DMS. Nevertheless, it does capture all inter-subsystem interactions. More components can be identified but they will be subcomponents of the already identified component list. The interactions of these subsystem subcomponents are either intra-component or are partial descriptions of the already identified interactions.

Gathering all of the DSM's interactions requires a user-friendly approach. To do this, two types of diagrams are proposed. Spatial and energy inter-subsystem interactions can be derived implicitly from a "material handling node diagram". Information interactions can be explicitly derived from a detailed hierarchical block control diagram. Methods for identifying intra-subsystem interactions can often be deduced at this stage of component aggregation. At much greater levels of detail, data such as CAD files, circuit diagrams, and UML diagrams can be relied upon.

3.3.1 Material Handling Node Diagram:

A material handling node diagram is composed of a set of nodes, a set of arcs that connect those nodes, and a set of labels to describe the arcs. A node is drawn for every discrete location and orientation in which a product is held stationary. These locations are usually equivalent to the set of independent and dependent buffers in the system. A dependent buffer is a manufacturing system artefact that is physically attached to any transforming machine or transporting material handler and is capable of storing raw material, WIP, or final goods at a specific location. In addition to buffers, material handler directors must also be included as nodes in the diagram. A one way arc between two nodes is drawn if there exists a material handler capable of moving from its end-effector (with or without product) between those two locations. The arc is labelled with all the material handlers and product parts that travel between those two points.

The labels of the material handling node diagram imply the spatial and energy interactions between subsystems.

0-7695-2589-X/06 $20.00 © 2006 IEEE

The product parts must arrive to the set of nodes and this requires spatial alignment between the part and the associated buffer. Once there, the buffer will have to hold the part; implying an energy transfer. These interactions appear directly in I_{LM}, I_{ML},I_{LM}, I_{LM} in the DSM. Similarly, the arrival and departure of a material handler to a node requires the spatial alignment to the associated buffers and material handling directors. These interactions appear in I_{MH}, I_{HM}, I_{HB}, I_{BH}, I_{HD}, and I_{DH}. Finally, the combination of part and material handler on an arc label implies spatial and structural interactions that appear in I_{LH}, and I_{HL}.

3.3.2 Detailed Hierarchical Block Control Diagram:

A detailed hierarchical block control diagram redraws the conceptual representation in Figure 1 as computational blocks that transfer information amongst each other. Acquiring this information can be potentially tedious in the absence of detailed I/O documentation. However, as a general rule, the physical parts of the system must communicate with their respective control components in the execution layer which in turn must communicate with their respective intelligent software in the planning and scheduling control layer. Communication between intelligent software agents is often document in the form of a UML sequence diagram. As of yet, no straightforward method has been found for identifying non-zero interactions within the execution layer.

4 HCBA Distributed Manufacturing System Example

A DSM analysis can now be carried out to motivate the application of the tool to distributed manufacturing systems. The robotic work cell, used in the assessment of the HCBA reference architecture is taken for study and shown in Figure 5. The system assembles a simple electrical meter box out of parts A, B, and C which are stored in an input and output buffer. The system is also composed of four manufacturing resources: a Hirata and Puma robot, a turn table and flipper to which each has its associated execution code and resource agents. A complete discussion of the cell can be found in [2]. This following subsystems are identified: L={Meter Box}, M={Hirata Robot}, H={Puma Robot, Flipper, Rotary Table}, B={Input Buffer, Output Buffer}. Their component lists are found in the first two columns of the DSM in Figure 8. Next, material handling node and detailed hierarchical block control diagrams are drawn in Figures 6 and 7. From these two diagrams, the system DMS in Figure 8 can be written straightforwardly. The HCBA robotic work cell design structure matrix immediately reveals a number of observations about the system. First, the product plays a strong coupling role. Not only must the product mechanical interface with all of the various tools, fixtures and end-effectors it encounters, but the product-agent also acts as a coordinator to the value-adding and material handling machines. Secondly, the integrative position of the rotary table and the PUMA robot appear as clear coupling to the remainder of the system resources. Finally, the DSM's white space shows the limited role of the Hirata robot and buffers. This example demonstrates that the DSM description of a system gives an immediate graphical clarification of the where coupling exists. The designer is immediately directed towards regions of high coupling so that she may act to reduce coupling if at all possible while maintaining the system's functionality.

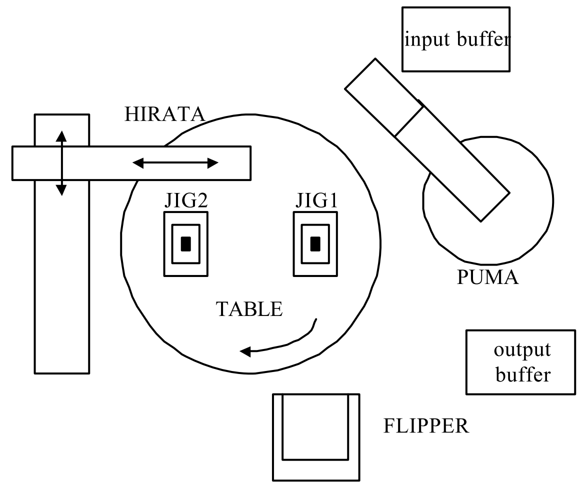

Figure 5. Schematic Diagram of the Robot Work Cell

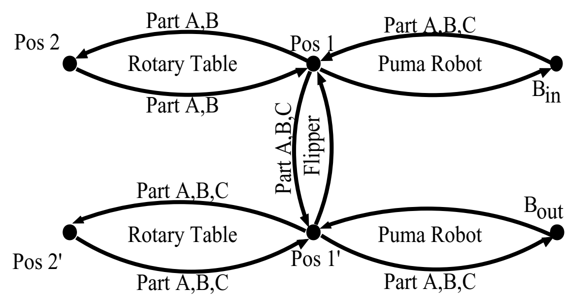

Figure 6. Robot Work Cell Material Handling Node Diagram

References

[1] C. Baldwin and K. Clark. Managing in an age of modularity. *Harvard Business Review*, (September-October 1997):84,

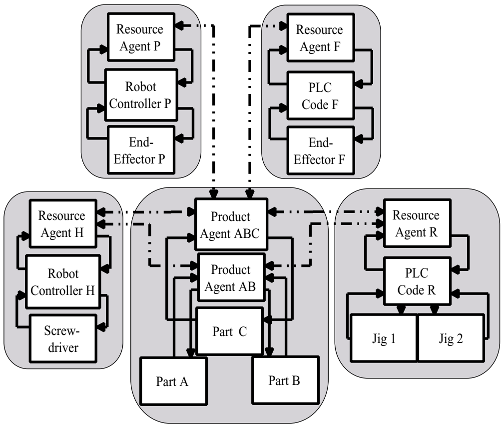

Figure 7. Robot Work Cell Detailed Hierarchical Block Control Diagram

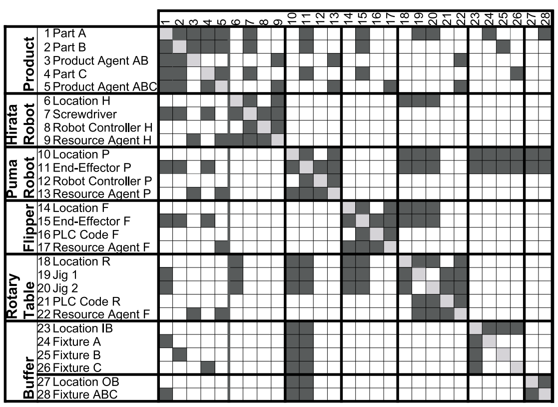

Figure 8. DSM of the Robot Work Cell

1997.

[2] J.-L. Chirn. *Developing a Reconfigurable Manuacturing Control System – A Holonic Component Based Approach.* PhD thesis, University of Cambridge, 2001.

[3] S. Eppinger. The design structure matrix - dsm. Technical report, http://www.dsmweb.org/, 2005.

[4] A. Farid and D. McFarlane. A tool for assessing reconfigurability of distributed manufacturing systems. In *submitted to INCOM 2006: 12th IFAC Symposium on Information Control Problems in Manufacturing*, Saint-Etienne, France, 2006.

[5] J. Gershenson, G. Prasad, and S. Allamneni. Modular product design: A life-cycle view. *Journal of Integrated Design and Process Science*, 3(4):13–26, 1999.

[6] J. Gershenson, G. Prasad, and Y. Zhang. Product modularity: definitions and benefits. *Journal of Engineering Design*, 14(3):295–313, 2004.

[7] J. Gershenson, G. Prasad, and Y. Zhang. Product modularity: measures and design methods. *Journal of Engineering Design*, 15(1):33–51, 2004.

[8] S. Goldman, R. Nagel, and K. Preiss. *Agile Competitors and Virtual Organisations*. Van Nostrand Reinhold, 1995.

[9] P. Gu, M. Hashemian, and S. Sosale. Integrated modular design methodology for life-cycle engineering. *CIRP Annals - Manufacturing Technology*, 46(1):71–74, 1997.

[10] C. Huang and A. Kusiak. Modularity in design of products and systems. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 28(1):66–77, 1998.

[11] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. Reconfigurable manufacturing systems. *CIRP Annals - Manufacturing Technology*, 48(2):527–540, 1999.

[12] D. McFarlane and S. Bussmann. Developments in holonic production planning and control. *Production Planning and Control*, 11(6):522–536, 2000.

[13] D. McFarlane and S. Bussmann. Holonic manufacturing control: Rationales, developments and open issues. In S. Deen, editor, *Agent-Based Manufacturing*, chapter 13, pages 303–326. Springer-Verlag, Berlin, 2003.

[14] D. McFarlane, S. Sarma, J. Chirn, K. Ashton, and C. Wong. Auto id systems and intelligent manufacturing control. *Engineering Applications of Artificial Intelligence*, 16(4):365–376, 2003.

[15] T. Pimmler and S. Eppinger. Integration analysis of product decompositions. In *American Society of Mechanical Engineers, Design Engineering Division (Publication) DE*, volume 68, pages 343–351, Minneapolis, MN, USA, 1994.

[16] R. Roberts. *Zone Logic - A Unique Method of Practical Artificial Intelligence*. Compute! Books, Radnor, PA, 1989.

[17] M. Sosa, S. Eppinger, and C. Rowles. Identifying modular and integrative systems and their impact on design team interactions. *Journal of Mechanical Design, Transactions of the ASME*, 125(2):240–252, 2003.

[18] S. Sosale, M. Hashemian, and P. Gu. Product modularization for reuse and recycling. In *American Society of Mechanical Engineers, Design Engineering Division (Publication) DE*, volume 94, pages 195–206, Dallas, TX, USA, 1997.

[19] D. Steward. Design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, EM-28(3):71–74, 1981.

[20] N. Suh. *Axiomatic Design: Advances and Applications*. Oxford University Press, New York, 2001.

[21] Y. Zhang, J. Gershenson, and S. Allamneni. Determining relationships between modularity and cost in product retirement. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 4, pages 49–61, Pittsburgh, PA, United States, 2001.

[22] Y. Zhang, J. Gershenson, and S. Allamneni. Determining relationships between modularity and cost in product retirement. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 4, pages 49–61, Pittsburgh, PA, United States, 2001.