A Design Structure Matrix Based Method for Reconfigurability Measurement of Distributed Manufacturing Systems

Amro M. Farid, Duncan C. McFarlane

(Invited Paper)

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 system; allowing it to readily adapt to changes over time. However, to date reconfigurability assessment has been limited, and hence the efficacy of the design approaches remains inconclusive. Recently, the "Design Structure Matrix" has been proposed as a tool for assessing the modularity of elements of a distributed manufacturing system and thereby providing an indirect indication of "reconfiguration ease"[16]. Additionally, an approach for its application has been proposed[14]. This paper develops this approach further into a systematic method for the reconfigurability measurement of manufacturing systems and illustrates its application on a robot assembly cell designed on distributed manufacturing system principles. This is achieved in three distinct phases: 1.) definition of system boundary 2.)decomposition of system functionality & components 3.)identification of component interfaces.**

Index Terms—**Reconfigurability, Reconfigurable Manufacturing Systems, Distributed Manufacturing System, Design Structure Matrix, DSM, Methodology**

I. 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[28][20]. In order to stay competitive, manufacturing firms have had to respond with high variety products of increasingly short product life cycle. 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[32].

Many approaches have been taken to try to achieve these dual requirements of mass-customisation and short product life cycle. Agile manufacturing systems developed in the 1990's to address every aspect of an enterprise's operations [20][33][23]. Agility, however, is primarily a business philosophy[30]. As a result, reconfigurable manufacturing systems arose to specifically address the ability with which production system's hardware and software could cope with frequent market change. A reconfigurable manufacturing system is defined as[24]:

A.M. Farid (amro.farid@cantab.net) and D.C. McFarlane (dcm@eng.cam.ac.uk) are with the University of Cambridge Institute for Manufacturing: 16 Mill Lane Cambridge CB2 1RX United Kingdom

The Cambridge University Institute for Manufacturing is part of the I*PROMS Network.

Definition 1.1: Reconfigurable Manufacturing System: "[A System] designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or regulatory requirements."

Reconfigurable manufacturing systems seek to achieve mass-customized and short life cycle products by incrementally adding capacity and functionality.

The structure (or architecture) of a manufacturing system must be considered in such a mass-customization, short product life cycle environment. The continual introduction of new product families and their associated variants requires that new production and material handling resources be easily added in order to adjust capacity and capability flexibly. Similarly, a new product introduction may require that the manufacturing system be rapidly redesigned in terms of a rearrangement of its production and material handling resources [24]. As demand for certain product variants ramp up, capacity can be installed incrementally. Finally, as demand for certain products falls off, the system can be reconfigured to support the potential growth of other product variants.

Assessing the suitability of a manufacturing system to these drivers requires measures of both its operation (behavioral) performances and its system (structural) performance. Measures for the former are well developed in the literature and industry. Among them are throughput, overall equipment effectiveness, lead time, etc. 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 [27].

This paper describes a method for the application of the design structure matrix as part of a larger framework to measure the reconfigurability of a manufacturing system based upon its structural properties[13]. Distributed manufacturing systems are taken as a specific class to which the general method may be applied. This is achieved with a background discussion of distributed manufacturing systems in Section II. Section III returns to the topic of reconfigurability and the requirements for its measurement. Section IV introduces the design structure matrix as a tool that partially fulfills these requirements. The paper then proceeds to its primary contribution with a method for the formulation of a production design structure matrix in Sections V and VI. This method is illustrated in Section VII in assessing a robot assembly cell designed on distributed manufacturing system principles.

II. Background: Distributed Manufacturing Systems

Distributed manufacturing systems have been identified as a key enabling technology in achieving the overall paradigm of reconfigurable manufacturing systemscite[24]. This section motivates further distribution in manufacturing control and then discusses how distributed manufacturing systems have been assessed in the context of reconfigurability.

A. 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 introductions may require that the manufacturing system be rapidly redesigned in terms of a rearrangement of its component production and material handling resources [24]. 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 [28].

To enable this rapid integration, a reconfigurable manufacturing system requires distributed or modular, open architecture controllers[24]. Each new resource, upon integration, requires a high level of autonomy which can be achieved by an additional distributed controller [8]. This autonomy means that decisions can be made locally without much affect on neighboring resources[10]. Furthermore, communication between controllers is limited to temporary and flexible relationships [3]. The resulting system is less complex[9] and more fault tolerant[4]. In these ways, distributed controllers improve system structure and behavior to facilitate the addition, change and removal of a new resource [8].

B. Definition & Scope

The subject of distributed manufacturing systems has been treated in a variety of fields at many different levels of scope. This paper restricts its discussion to shop-floor activities and defines a distributed manufacturing system (DMS) as:

Definition 2.1: Distributed Manufacturing System: a system that uses a collection of value-adding and material-handling resources which are controlled by a DMS control system to transform raw material into finished product.

Furthermore a DMS control system is:

Definition 2.2: DMS control system: 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.

These decision elements may be broadly classified as intelligent software/agents for high level decision making, and automation objects for the execution functionality. A distributed manufacturing system may also use the part-oriented control[17] or intelligent product[29] paradigms. A conceptual representation of a distributed manufacturing system is shown in Figure 1.

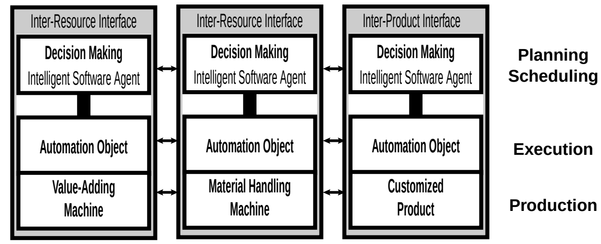


Fig. 1: A Conceptual Representation of a Distributed Manufacturing System

C. Assessment Techniques

There has been a significant number of distributed manufacturing systems introduced in the literature which comply with definition in Section II-B. Of these, three, PROSA[2][39], ADACOR[26] and HCBA[6] have been designed as general reference architectures, and later implemented into specific cases as system architectures. As such, they provide suitable examples for a discussion on distributed manufacturing system assessment techniques. PROSA's evaluation method was primarily qualitative. Evaluation techniques from the architectures of buildings and object-oriented software were borrowed in order to discuss descriptively the adherence of the architecture to the identified design requirements [39]. The evaluation method also relied on the flexibility of the architecture's associated algorithms [2]. ADACOR's evaluation technique measured operational performance measures such as throughput and lead time under various disturbance scenarios as a function of varying architectures: hierarchical, heterarchical and hybrid (ADACOR) [26]. Here, the comparison could be made due to the adaptive nature of the ADACOR architecture because it used an algorithm similar to the baseline hierarchical and heterarchical architectures. HCBA used structural measures such as petri-net complexity and lines of code. These measures were then used to calculate extension and reuse rates for various reconfigurations such as the addition of new machines[6].

Although instructive, these evaluation techniques indicate a lack of existing reconfigurability measurement techniques. In the case of the first two reference architectures discussed, (PROSA and ADACOR), evaluation was carried out either qualitatively or quantitatively by measuring operation (behavioral) performance. The assessment of the last of the architectures, HCBA, added to the evaluation literature by proposing structural measures. However, the relationship of these metrics to reconfigurability needs to be clarified in order to make conclusive statements about reconfigurability improvements.

III. Reconfigurability: A Property of Manufacturing Systems

Having overviewed distributed manufacturing system as a specific class of manufacturing systems, the discussion returns to the development of a general method of assessing reconfigurability in manufacturing systems. This section seeks to illuminate reconfigurability as a property of manufacturing systems. From this, a set of requirements can be identified for its measurement.

A. Reconfigurability Definition

Throughout the literature, many definitions for reconfigurability have been proposed. Two indicative definitions are "the ability to repeatedly change and rearrange the components of a system in a cost-effective way"[34] and "the ability of a function of a manufacturing unit to be simply altered in a timely and cost effective manner " [28]. The first treats reconfigurability purely in terms of system components. This, however, does not explicitly address the arrangement of functions that affect manufacturing processes. Conversely, the latter does not explicitly address the need to rearrange components in order to realize a rearrangement of manufacturing functions. Both of these definitions also do not explicitly state that not all reconfigurations are desirable. The discussion of reconfigurability originates from a need to introduce new product variants and manufacturing resources in such a way that system capacity and capability closely match the breadth of the product line. In this way, one must introduce the notion of a set of potentially desired alternate configurations such as the addition of new products and/or machines. To support these issues, the following encompassing definition is proposed:

Definition 3.1: Reconfigurability: The ability to add, remove and/or rearrange in a timely and cost-effective manner the components and functions of a system which result in a desired set of reconfigurations.

B. Manufacturing Reconfigurability Characteristics

The reconfigurability of a manufacturing system can be further understood in terms of five characteristics it exhibits [30]:

- Modularity: The degree to which all system components, both software and hardware are modular.
- Integrability: The ability with which systems and components maybe readily integrated and future technology introduced.
- Convertibility: The ability of the system to quickly changeover between existing products and adapt to future products.
- Diagnosability: The ability to quickly identify the sources of quality and reliability problems that occur in large systems.
- Customization: The degree to which the capability and flexibility of the manufacturing system hardware and controls match the application (product family).

These characteristics emphasize that the capacity and functionality of a reconfigurable manufacturing system change over time so as to both diversify the product line and to match the capacity to the demanded quantity. To achieve the extensible nature of reconfigurability and its key characteristics, open, modular architectures are required at hardware, control, and software levels. These in turn require well-defined interfaces without which any reconfiguration process would be both lengthy and costly [24].

C. Requirements for Reconfigurability Measurement

From the understanding of reconfigurability developed in the previous two sections, a set of requirements can be developed for its measurement. Broadly speaking, these can be divided into three categories 1.)requirements for measurement 2.)requirements for describing reconfigurability 3.)suitability requirements for manufacturing systems.

1) Requirements for Measurement: The process of measurement generally has five requirements:

1) Identification of structure dependent measurables
2) Methods for measuring the measureables
3) Models for describing/modeling system
4) Identification of structure dependent properties
5) Formulaic measures of relating those models to those properties

The requirements can be graphically represented as a sequential data flow diagram in Figure 2. Implicitly, as an

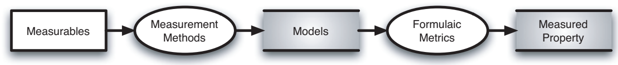


Fig. 2: A Generic Measurement Process

objective, a set of measured properties need to be identified. In most measurement processes, these properties are distinct from the system measureables which must be extracted from the system of interest with a set of measurement methods. If the measured property is too complex for direct measurement, the measurement must be inferred [5]. This requires that the measureables be related using a set of models. Finally, the mathematical theory of measurement [7] requires a set of measures. These measures are a specific class of mathematical functions and serve to convert related measureables to the final measured property [11].

2) Requirements for Reconfigurability Description: In addition to the general requirements of measurement, requirements for describing reconfigurability need to be added. From the proposed definition given in Section III-A four pieces of information are required to describe reconfigurability.

1) Definition of system and its boundaries
2) Description of system configuration: functions, components & their interrelationships
3) Description & rationale for desired set of reconfigurations
4) Description of time and/or cost of potential reconfigurations

First, any description of a reconfigurable system implicitly requires that the system and its boundaries be defined. While this may seem obvious, a reconfigurable system provides a unique challenge in that its definition may change over time. To overcome this, a reconfigurable system is analyzed at a particular instant in time prior to a reconfiguration process. Next, the system configuration must be described in terms of its functions, components, and their inter-relationships. Two types of relationships can be studied: function-component relationships and component-component relationships. The former describes the allocation of functionality to system components and hence gives a measure of its capabilities [15].

The latter describes the component-component interfaces previously identified as necessary for the effective realization of reconfigurable manufacturing systems. While, this requirement is also intuitive, its fulfillment is complicated by the choice of granularity of the description. This difficulty, however, is partially mitigated by describing the set of desired configurations and the rationale for their existence. Components involved in a potential reconfiguration would require greater description than those for which no reconfiguration process is foreseen. Finally, to assure the timeliness and cost-effectiveness of potential reconfigurations, the reconfigurability measure would require some estimation of reconfiguration time and cost.

3) Suitability for Manufacturing Systems: In addition to the general measurement requirements for the reconfigurability of a system, further requirements are necessary due to the special characteristics of a manufacturing system. In this regard, a tailored reconfigurability measure must necessarily address the transformation and transportation processes of a manufacturing system and the components/resources that realize them. These processes and their sequence may occur over multiple energy domains. Therefore, any models used must be rich enough to describe the diversity of mechanical, electrical, chemical and information processes. Similarly, interfaces may exchange material, energy and/or information. Finally, the models used must accommodate the broad heterogeneity of technologies often used in manufacturing.

IV. Modularity Assessment using the Design Structure Matrix

The previous section described the definition, characteristics, and requirements for reconfigurability and its measurement. In so doing, it described the need for system structure models upon which formulaic techniques can be used to yield reconfigurability measures. This section proposes the so called design structure matrix (DSM) as such a modeling tool[12][37]. In particular, it shows promise in assessing the modularity of reconfigurable manufacturing systems. The section is divided into two parts. First, the DSM is introduced with a brief description. Second, the various usages of the DSM are reviewed.

A. Description of the Design Structure Matrix

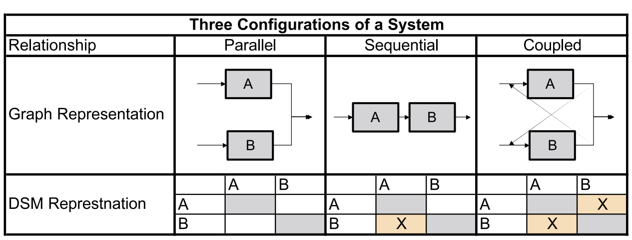


Fig. 3: DSM Representations of System Configurations

The design structure matrix is a systems analysis tool that captures the interactions, interdependencies, and interfaces between components of a complex system in a compact and clear representation [12]. Given two components A and B, they may interact in a parallel, sequential or coupled fashion. These interactions may be spatial, structural, energy, material or information interfaces [35]. Figure 3 shows the graphical representation of these interactions and their associated design structure matrices. Essentially, off-diagonal elements reflect structural interaction. The placement of an off-diagonal "X" represents the existence of an interaction between two components A and B [12]. Some authors, however, have replaced the "X" with numerical values in order to subjectively assess the strength of a particular interaction [31][40].

B. Usage of the DSM: An Overview

The DSM has found many uses in the field of product design. Within the scope of this discussion, the most relevant of these is 1.) the modeling of the system structure 2.) calculating the modularity of that system. Pimmler and Eppinger used the DSM to model the structure of an automotive climate control system and then used the analysis to advance concepts in the modularity of subsystems [31]. Similarly, Sosa et al. used the DSM to analyze the interactions of a large commercial aircraft engine. The analysis was used to advance a methodology of allocating design teams to major aircraft subsystems [35]. In this latter case, one can draw an analogy between the transportive and transforming functions of an aircraft engine to those of manufacturing systems. Both systems also require many layers of control and are similarly complex.

The DSM has also served as a data structure from which a variety of modularity measures have been developed. They generally use off-diagonal summations of the DSM but disagree on the exact formulaic description depending on application, and underlying assumptions. Gershenson has conducted an exhaustive review of these measures [19] and their associated definitions [18]. One particularly interesting modularity measure is the so called group efficacy metric [25].

$$\mathcal{M} = \frac{e_d}{e_b + e_o} \tag{1}$$

where e_d is number of full elements in the block diagonal, e_o is the number of full elements in the off-block diagonal, and e_b is total size of the block diagonal. This metric has a number of useful intrinsic features such as a meaningful denominator and extrema[25].

Interestingly, there is much similarity between modularity applications in the field of product design and reconfigurability applications of manufacturing systems. Huang and Kusiak have discussed matrix-based modularity measures to facilitate the realization of highly customized products [22]. The product modularity necessary to achieve customization appears to correspond to the manufacturing system modularity necessary to achieve reconfigurability. In one sense, the production of a modular product is facilitated by a manufacturing system designed along modularity principles. In another sense, a reconfigurable manufacturing system is a system that undergoes customization over time much like a customized product line. Matrix-based modularity measures have also been used to advance the role of modularity in life cycle engineering [21][36][40]. Analogously, modularity may play a role in the efficient operation, maintenance, and decommissioning of manufacturing systems.

V. Production DSM Formulation

Having described the fundamentals of the DSM, the next two sections shift focus to its application in a production environment. In this section, the production DSM is formulated in three steps: 1.) definition of system boundary & functionality 2.) decomposition of system functionality & components 3.)identification of component interfaces.

A. *Definition of System Boundary & Functionality*

In Section III-C2, the definition of a system's boundary and functionality was identified as a requirement for a reconfigurability description. However, design structure matrices typically analyze closed systems and so an inherently open manufacturing system must be translated to an analogous closed system. As illustrated in Figure 4, manufacturing systems convert information, energy, and material into other forms of information, waste energy, and material. The production 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 and an infinite sink to the generated outputs. Good examples of such sources and sinks include orders and their corresponding fulfillment data. The system boundary also serves as a common platform to which all the manufacturing system components can connect. For example, basic factory services such as power and networking are included as part of the system boundary. 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.

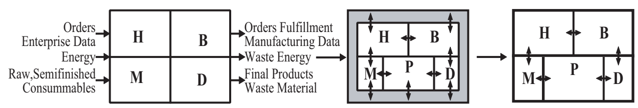


Fig. 4: Analogous Open and Closed Manufacturing Systems

B. *Decomposition of System Functionality & Components*

The identification of the subsystem components is not trivial and more than one set of component aggregations can be conceived to describe a given manufacturing system. One approach to identifying the components is to use Axiomatic Design Theory for large flexible systems[38]. Specifically, the functional requirements are a set of transformation, transportation, and storage processes. They can be allocated flexibly to the transforming machines, material handlers and buffers. These production processes can be further decomposed into sub-functions which must have their corresponding subsystem components. Figure 5 illustrates the axiomatic design theory approach. In a complementary approach, Baldwin & Clark identify components based upon the principle of visible design rules[1]. Using this strategy, subsystem components can be identified based upon the clear interfaces between them. A combination of these two approaches is used here.

Using this approach, three general functional requirements are identified for a manufacturing systems 1.)transform products 2.)transport products and 3.) store products which may be allocated to their respective subsystems.

$M = \{m_1 \ldots m_{\sigma(M)}\}$ – A set of $\sigma(M)$ (value-adding) machines capable of realizing one or more transformations on the products of the product line. The $\sigma()$ operator gives the size of a set.

$H = \{h_1 \ldots h_{\sigma(H)}\}$ – A set of $\sigma(H)$ material handlers capable of transporting raw material, WIP, and/or final goods in the product line between a given pair of value-adding machines and/or independent buffers

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

Additionally, as mentioned in the previous section, the manufacturing system's products must be included as subsystems.

$L = \{l_1 \ldots l_{\sigma(L)}\}$ – A product line composed of $\sigma(L)$ products. Depending on the application, these products may adhere to the intelligent product paradigm[29].

Each of these subsystem can be decomposed further into components. Subsystems m_i, h_i, b_i and l_i have component sets C_{m_i}, C_{h_i}, C_{b_i} and C_{l_i} respectively.

*Transforming Machine Components:*A machine must have a tool and a fixture to form and hold the product respectively. These two components may be simple, or they may be treated as aggregations with their own set of subordinate components. For example, a machine may be composed of complex fixturing and tooling systems that flexibly allow for multiple configurations of tools and fixtures. Additionally, the machine must have control components. These can include controllers devoted to continuous real-time, execution, scheduling or planning. 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.

$$C_{m_i} = \{\text{Location, Tool(s), Fixture(s), Controllers}\} \quad (2)$$

*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 controllers devoted to continuous real-time, execution, scheduling or planning. Implicitly, the machine must also have a region of motion by which to relate itself spatially to the other manufacturing subsystems.

$$C_{h_i} = \{\text{Motion Region, End-Effectors(s), Controllers}\} \quad (3)$$

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

$$C_{b_i} = \{\text{Location, Fixture(s), Controllers}\} \quad (4)$$

Product Components: Product components can be as simple as a bill of material for assembled products or product features for non-assembled ones. Such features may include slots, holes or chamfers. Also, given intelligent products or part-oriented

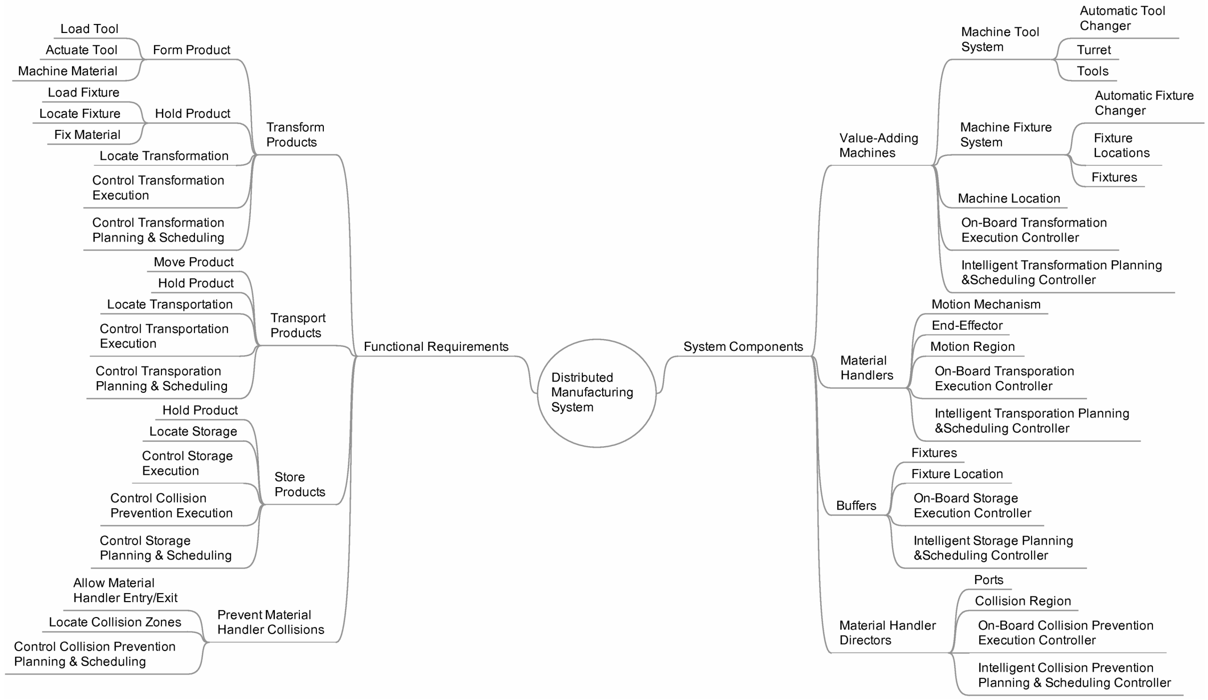


Fig. 5: Axiomatic Design of a Generic Distributed Manufacturing System

control, a number of intelligent software components will be needed to control the planning and scheduling activities of the various subassemblies.

$$C_{l_i} = \{\text{Parts/Features, Intelligent Software(s)}\}. \quad (5)$$

C. Identification of Component Interfaces

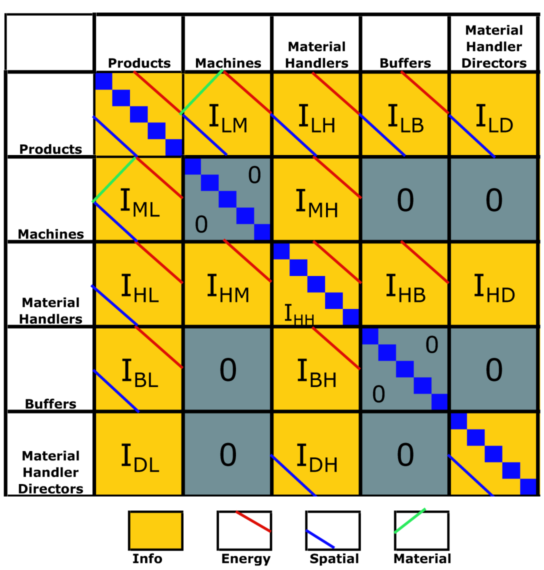


Fig. 6: Distributed Production Design Structure Matrix

Having defined the manufacturing system components, the DSM is constructed, as shown in Figure 6. Capturing all possible component-component interactions is a daunting process even for modestly sized systems. The size of the DSM grows quickly with the number of components and testing the existence of an interface between any pair of components is both tedious and time consuming. To overcome this challenge, typical component-component interactions can be identified and classified a priori. First, the matrix elements where no interaction exists can be identified. In this way, effort can be focused on the nonzero interactions. Typically, no $M-B$, $M-D$, $B-D$ interactions seem to exist. Additionally, the interactions in the I_{LL}, I_{MM}, I_{BB} 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_{HH} must allow spatial interactions between two material handlers.

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 plentiful and may occur within subsystems and also between any subsystem and a material handler, or product. The specifics of these interactions is highly dependent on the control architecture. 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.

VI. Application to Industrial Study

The previous section formulated the production DSM in terms of a production system's subsystems, constituent components and the interfaces between them. This section builds on this formulation with an industrial method that extracts these measurables from the manufacturing system and enters them into the DSM. First, the challenges of manual usage of the DSM are discussed and then a case-study based method is overviewed.

A. Challenges of Manual Usage of the DSM

The DSM as an analytical tool presents a number of problems to the industrial engineering analyst. The number of matrix elements grows as the square of the number of components. This means that the analyst expends an ever increasing amount of analytical effort in order to achieve incremental improvements in accuracy. Furthermore, the tedious and slow nature of filling each element with a binary value could lead to many errors. Even if these problems are overcome, the analyst must use a systematic approach of considering each matrix element. Without it, missing or double counting would be very easy to miss or double count interactions.

Many of these difficulties can be overcome using an automated analytical technique. However, in the absence of such a technique, one may resort to a manual case-study based method. Unfortunately, manual DSM analysis has some inherent human factors challenges. It requires that the analyst simultaneously maintain both detailed interface information and broad information across the entire scope of the manufacturing system. In such a situation, the analyst is likely to become entrenched in the detailed study of component interfaces without gaining an appreciation for the system as a whole. Simultaneously maintaining both broad and detailed information in manual case study investigation is a great challenge.

To overcome this challenge, two strategies are used in the development of a three phased manual case study based method. Firstly, the method utilizes an "outside-in" analytic approach. In other words, once the system boundary is delineated, this boundary can be used to more easily identify the interfaces of subsystems, and then once again in order to identify the interfaces of components. Secondly, the method uses a "decremental-scope incremental-detail" approach. The clarification of inter-component interface is taken as an iterative process in which once an interface is identified, it is detailed in a series of rough to fine passes.

B. Case Study Based Method: Overview

These strategies were used to develop a three-phased case study based method. The first phase consists of an introductory visit to the production facility in which the analyst gains a high level overview of the production system. The second phase consists of a second visit to the facility; this time for the collection of detailed configuration data. In the final phase, this data is aggregated and analyzed in an almost entirely automatic fashion to generate the production DSM. A schematic overview of the case study method is shown in Figure **??** and more detailed explanation is given in [13].

Each of these phases produces its corresponding data structures. As an orientation visit, the first phase produces four data structures:

1) the set of desired reconfigurations
2) the production system boundary and functionality
3) a plant layout diagram
4) a simple hierarchical block control diagram

The set of desired reconfigurations is required by the reconfigurability definition and gives sense to detailed decomposition efforts done in later phases. The production system functionality and boundary are required as part of the formulation described in the previous section. A plant layout facilitates the identification of spatial and structural interactions. Finally, a simple hierarchical control diagram is required as an incremental step to the later generation of the detailed hierarchical block control diagram. Both of these diagrams are later described in Section VI-C.

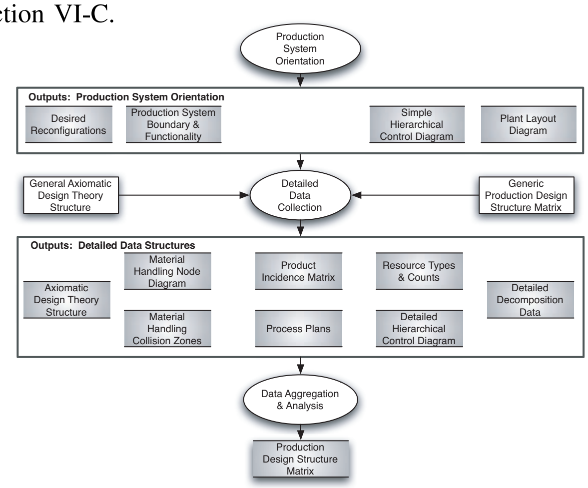


Fig. 7: Schematic Overview of Case-Study Based Method

Phase 2 takes the data structures produced in Phase 1 and adds to them the general axiomatic design theory and DSM structures shown in Figure 5 and 6. These two data structure can be viewed as going through an instantiation process during Phase 2's detailed visit. To facilitate this instantiation process, the analyst may use a set of questionnaire forms[13] which describe the components and interfaces of the six types of subsystems in tabular form. The result of the Phase 2 effort is eight different data structures and diagrams. This large number is motivated by the dual need to both acquire information in readily available forms during the case study, and to address specific parts of the DSM at one time. Each of these structures are briefly described in Section VI-C.

From these data structures, the production DSM can be generated automatically in Phase 3. MATLAB software was developed for this purpose. In this way, general information achieved in Phase 1 was detailed into a number of easily acquired and manageable data structures. These data structures have the added benefit that they are in a form which can be easily parsed and aggregated automatically to form the DSM.

C. Description of Detailed Visit Data Structures

Each of the data structures acquired in the detailed visit are now overviewed.

1) Axiomatic Design Theory Structure: In order to determine the component labels of the DSM, the general axiomatic design theory structure shown in Figure 5 is instantiated during the detailed visit. This provides a straightforward way to identify the five types of subsystems. Also, because the general structure already identified common subsystem components, this effort does not need to be expended for each new subsystem encountered in the study. Finally, the tree shape provides a convenient structure from which to decompose the system components further.

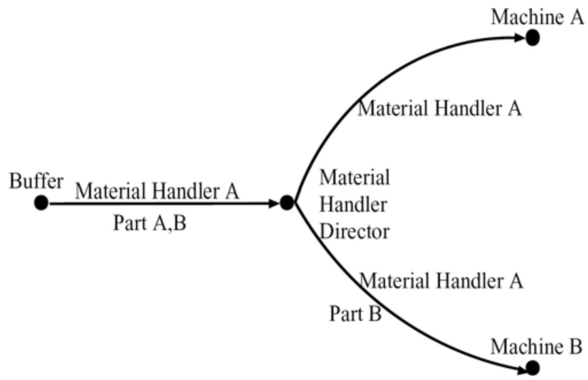


Fig. 8: A Generic Material Handling Node Diagram

2) 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. A labelled one way arc between two nodes is drawn if there exists a material handler capable of moving its end-effector (with or without product) between those two locations. A generic example of a material handling node diagram is shown in Figure 8. The material handling node diagram implies the spatial and energy interactions which appear in I_{MH}, I_{HM}, I_{HB}, I_{BH}, I_{HD}, and I_{DH}.

3) Material Handling Overlap Zones Table: In any given manufacturing system, there may exist spatial regions through which more than one material handler may pass. As the motion region of a material handler is one of its components, this causes spatial interaction and appears in I_{HH}. To capture this type of interaction easily, Table I can be used.

TABLE I: Material Handler Overlap Zones

Material Handler Shared Space	**Name**
Material Handler 1	Name
Material Handler #	Name

4) Product-Resource Incidence Matrix: The Product-Resource Incidence Matrix is another intuitive data structure that is often readily available in production facilities. This matrix simply states if a given product receives a transformation, transportation, and storage process during the course of its manufacture. To receive these operations, the product requires spatial alignment and energy interactions between it and the associated resources. These interactions appear in I_{LM}, I_{ML}, I_{LH}, I_{HL}, I_{LB}, and I_{BL}. A generic product-resource incidence matrix is shown in Table II.

TABLE II: Product-Resource Incidence Matrix

	Machine	Material Handler	Buffer
Product 1	X		X
Product 2	X	X	
Product #		X	X

5) Resource Types & Counts Table: The resource types and counts ensures that the number and type of each resource is explicitly written. A generic resource types and counts table is shown in Table III.

TABLE III: Resource Types & Counts Table

Resource Name	Type	Count
Milling Machine	Machine	2
Robot #	Material Handler	1

6) Process Plans: Process plans are another example of intuitive data that is readily available at many production facilities. It simply describes for each product component or feature the name of each transformation process it requires and on which machine this transformation occurs. These interactions appear in the I_{LM} and I_{ML} matrices. Table IV shows a generic process plan.

TABLE IV: An Example Process Plan

	Product	Name	
Op. #	Feature/Component	Process	Machine
1	Component 1	Milling	Milling Machine
2	Assembly	Assembly	Robot

7) Hierarchical Block Control Diagrams:: The hierarchical block control diagrams draw the control elements of the manufacturing system as computational blocks that transfer information amongst each other. The difference between the simple and detailed versions is that the simple version captures only the different types of controllers in the manufacturing system while the detailed version shows each resource and its corresponding control elements. These information interactions contribute to the DSM in all places where Figure 6 previously identified the potential for information interactions. A generic hierarchical control diagram is shown in Figure 9. Acquiring this information, however, can be potentially tedious in the absence of detailed I/O documentation. A bottom-up approach is suitable for the analysis of the manufacturing system control structures. After each subsystem has been identified, the analyst can determine the lowest level of control for each production resource. Upon completion, she may proceed to the next level taking into account centralized elements where necessary.

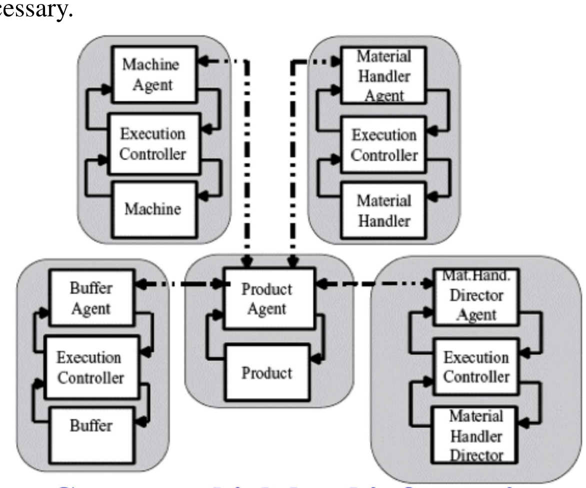


Fig. 9: Generic Hierarchical Control Diagram

8) Detailed Decomposition Data: In addition to the previous seven data structures further insight can be gained by decomposing components into subcomponents and addressing their interfaces in greater detail. Decomposition of all components is not necessary. Rather, this detailed effort may be localized to subsystems and components which may undergo one of the desired reconfigurations.

Detailed interface data comes in many forms depending on the type of interface: information, energy, material or spatial. It also depends on the available documentation for the manufacturing system. Spatial and structural interaction are often captured in CAD data. CAM software is a good source of energy interactions between a product and value-adding machines. Information interactions may come from circuit diagrams, control block diagrams, or UML diagrams depending on the applicable control level. Other sources of detailed interface data may exist and it is up to the analyst to determine them for any given potential reconfiguration.

VII. A Distributed Manufacturing System Example

A simplified DSM analysis can now be carried out to motivate the application of the method. The robotic work cell, used in the assessment of the HCBA reference architecture, shown in Figure 10, is taken for study. 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 a flipper to which each has its associated execution code and resource agents. A complete discussion of the cell can be found in [6].

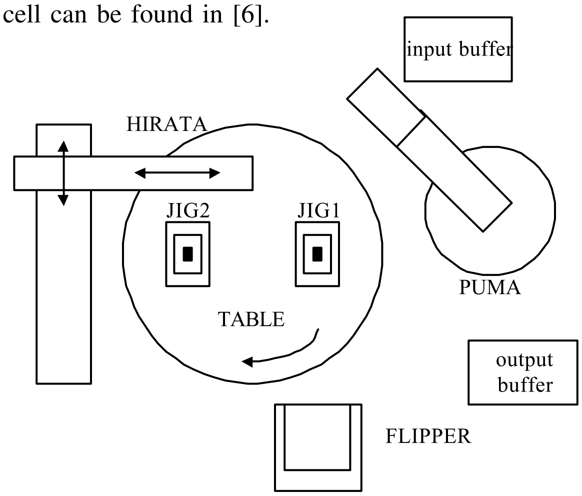


Fig. 10: A Schematic Diagram of the HCBA Robot Work Cell

A. Definition of System Boundary & Functionality

In this example, the definition of the system boundary is straightforward. The input and output buffers form natural boundaries on the flow of material. Energy flows in and out of the system such as electrical power and heat are modelled as infinite sources or sinks that the boundary provides. Normally, the cell required user input in the form of orders. It also notified the user of order completion. To complete the translation of this open system into a closed one, these information flows are modelled as terminating at Product Agent ABC.

B. Decomposition of System Functionality & Components

The decomposition of the system functionality and components is similarly straightforward. The distributed manufacturing system axiomatic design theory structure provided in Figure 5 is instantiated for each type of subsystem. The 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 full axiomatic design theory structure in Figure 11.

C. Identification of Component Interfaces

Although component interfaces can be identified manually for such a modestly sized system, the tools described in Section VI-C are used in order to demonstrate the method and the developed software. The robotic work cell material

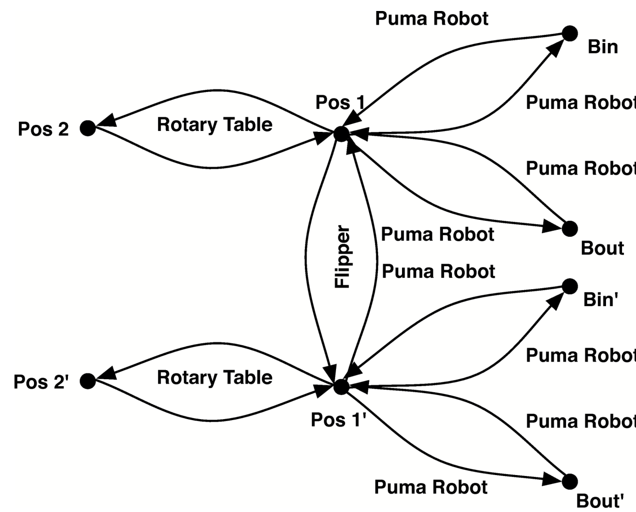


Fig. 12: Robot Work Cell Material Handling Node Diagram

handling node diagram shown in Figure 12 has eight nodes. Normally, the system would have only four nodes but as part orientation is critical each node must have its opposite orientation. These nodes are connected with arcs wherever there is a material handler capable of realizing the transportation process. Interestingly, this assembly cell does not make use of all the nodes and arcs in the diagram, as not all are needed for the assembly of the meter box.

The previously described tabular tools are also straight forward in this example system. The Puma robot, rotary table and the flipper spatially overlap at Jig 1. In order to be fully assembled, Meter Box ABC must interact with all resources of which there is only one of each kind. Finally, the process plan describes the two assembly operations which the Hirata robot completes.

TABLE V: Material Handler Overlap Zones

Material Handler Shared Space	Name
Material Handler 1	Puma Robot
Material Handler 2	Rotary Table
Material Handler 3	Flipper

The detailed hierarchical control diagram provides an intuitive picture of a part-oriented control distributed manufacturing system. Each subsystem has its associated execution code and agent. The diagram clarifies the communication between agents. In this case, all coordination is done through the two product agents. It also shows the absence of any inter-subsystem communication at the execution level.

TABLE VI: Product-Resource Incidence Matrix

	Product ABC
Hirata Robot	X
Puma Robot	X
Flipper	X
Rotary Table	X
Input Buffer	X
Output Buffer	X

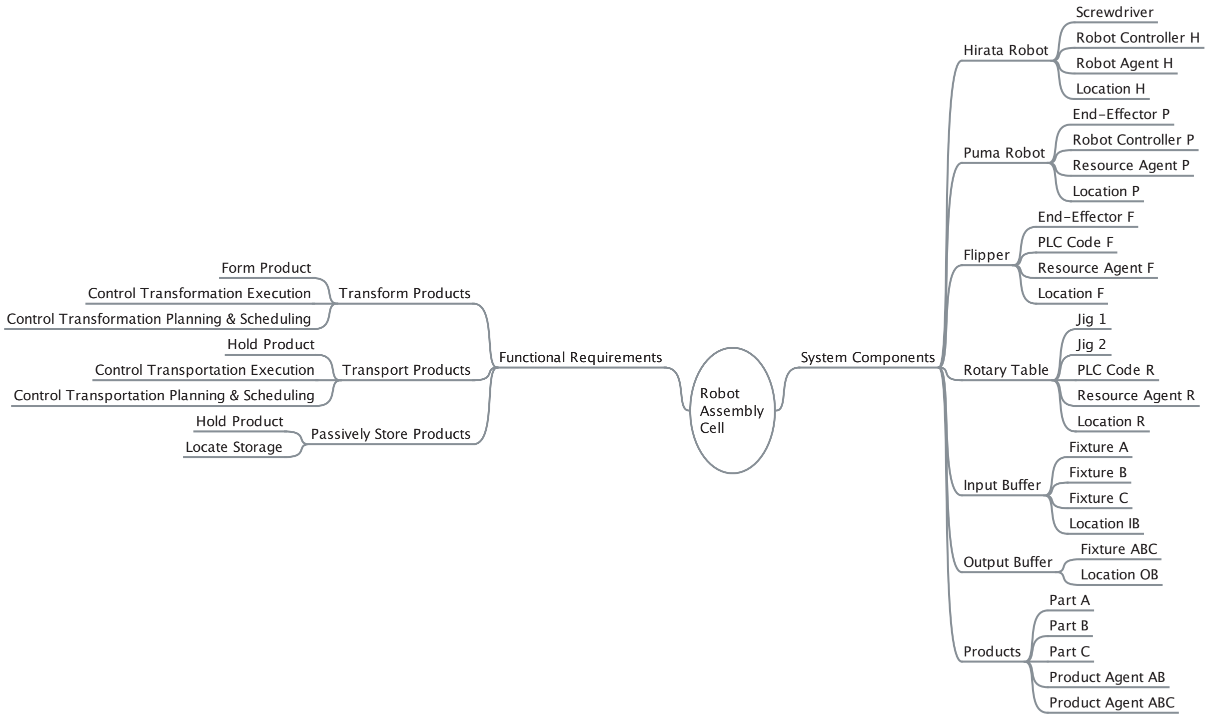


Fig. 11: Axiomatic Design of Robot Assembly Cell

TABLE VII: Example of Resource Types & Counts

Resource Name	Type	Count
Hirata Robot	Machine	1
Puma Robot	Material Handler	1
Flipper	Material Handler	1
Rotary Table	Material Handler	1
Input Buffer	Buffer	1
Output Buffer	Buffer	1

TABLE VIII: An Example Process Plan

	Product	Name	
Op. #	**Feature/Component**	**Process**	**Machine**
1	Part A & B	Assembly	Hirata Robot
2	Subassembly AB & Part C	Assembly	Hirata Robot

1) Detailed Decomposition Data: At this point, the production DSM can be constructed as shown in Figure 14. However, detailed decomposition data can be added to expose the complexity of inter-subsystem interfaces. In the case of the design and evaluation of distributed manufacturing systems, the inter-agent communication is of prime importance. The UML sequence diagram of the multi-agent system seen in Figure 15 can be added to the production DSM. In this way, the design of the distributed control algorithm can be evaluated in the context of the overall system modularity.

D. Production Design Structure Matrix Results

The results of the production DSM can be discussed at both a general and a detailed level. Figure 14 reveals the coupling among the system's components. The product's coupling is most prominent. Not only must the product mechanically interface with all of the various tools, fixtures, ad end-effectors it encounters, but its agents also coordinate product information to all of the value-adding and material handling resources. A second observation is the degree of heavy coupling along the block diagonal. The integrative role of material-handling resources such as the rotary table and PUMA robot is also

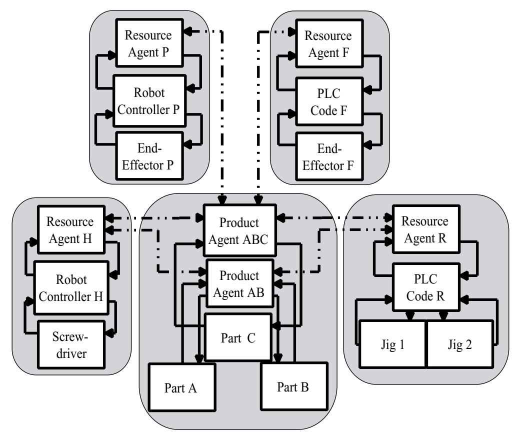


Fig. 13: Generic Hierarchical Control Diagram

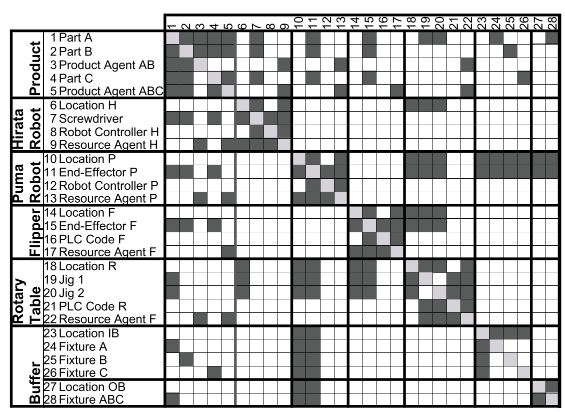


Fig. 14: High Level Production Design Structure Matrix

noted. Finally, the tertiary role of the buffers and the Hirata robot appears as DSM white space. Finally, the existence of white space in the DSM is as descriptive as marked elements. Here, the tertiary role of the buffers and the limited role of the

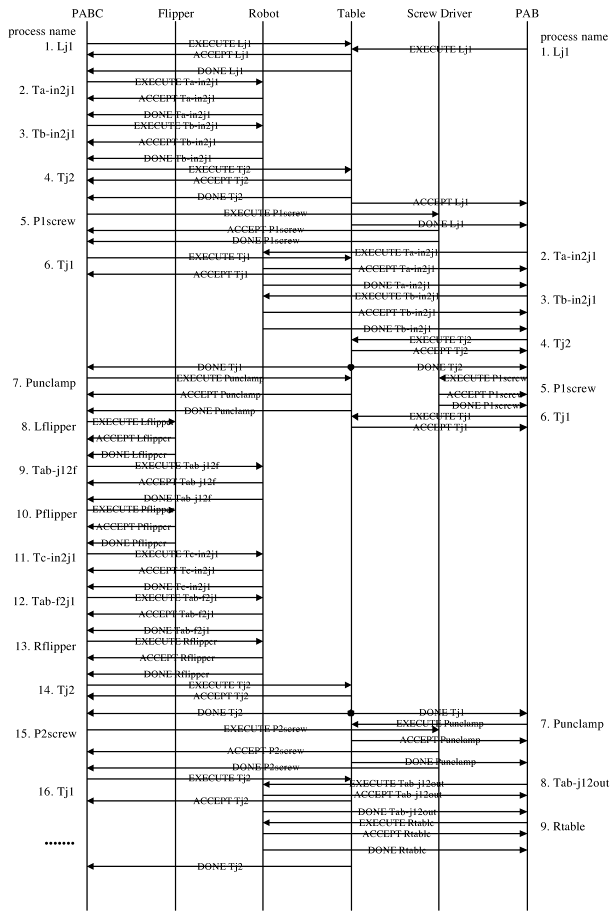


Fig. 15: Agent Sequence Diagram

Hirata robot becomes clear. This example demonstrates that the DSM directs the designer towards regions of high coupling so that she may act to reduce coupling while maintaining the system's functionality.

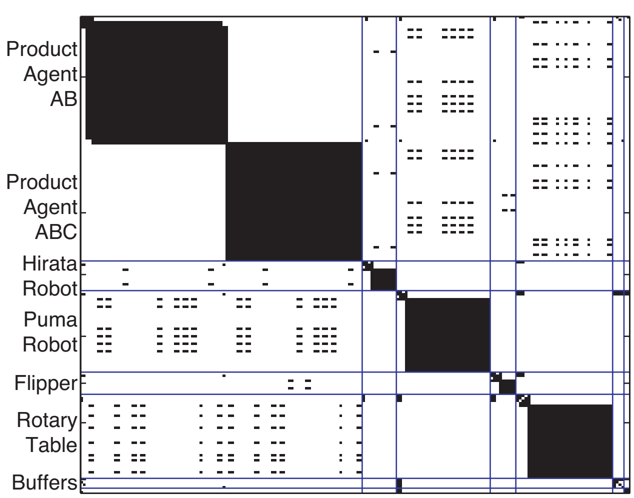


Fig. 16: Detailed Production Design Structure Matrix

The DSM can also be an effective tool at a more detailed stage of design and evaluation. Using the UML sequence diagram found in Figure 15 as detailed decomposition data, the DSM is expanded significantly in size to the one shown in Figure 16. The resulting DSM shows the coupling in the production system's planning and scheduling control layer. At a detailed level, the block diagonal takes on an even more prominent role. Its perfectly filled blocks arise from an assumption that intra-agent interactions were fully integrated. The size of each block on the diagonal is an indication of the complexity and functionality of each subsystem agent. The two product agents play an extensive coordinating role which often requires communication with material handlers like the Puma Robot and the Rotary table. At the other extreme, the Hirata robot plays a minimal role in the agent interaction and coordination while the two buffers practically disappear. In this way, the DSM can be used as a detailed tool that can be focused on the most pertinent aspects of the system.

Finally, the production DSM also allows for quantitative assessment using the group efficacy metric given in Equation 1 for modularity measurement.

$$\mathcal{M} = \frac{e_d}{e_b + e_o} = \frac{6800}{12291 + 728} = 0.5223 \qquad (6)$$

This number alone provides little insight as modularity is agreed by most research to be a relative rather than an absolute metric[19]. However, the values serves as the basis for improvement in later designs and/or reconfigurations. Off-diagonal elements can be removed or intra-subsystem interactions can be strengthened. It is worthy of mention, that the choice of subsystem boundaries plays a critical role in the modularity value. In this example, the meter box was treated as one product. As a result, its e_d/e_b value was significantly diminished. Given, the importance of the product in this system, an analysis that identified two product systems would have yielded significantly higher modularity results.

VIII. Conclusions & Future Work

This paper has proposed the design structure matrix as a model in a larger indirect measurement process. At the same time, the tool showed its suitability to manufacturing systems because it was able to equally treat heterogeneous technologies, processes in multiple energy domains, and all types of component interfaces. The model also facilitates quantitative assessment through metrics like group efficacy.

This paper has also described a user-friendly method for the application of the DSM in a case study environment. Using the DSM manually provides many challenges. This method uses common features of manufacturing systems to orient the analyst at the beginning of a measurement process. Additionally, the method employs a three phase approach to construct the DSM in an intuitive and incremental fashion. Finally, the method automates the data aggregation and analysis so as to diminish the potential for errors in repetitive time consuming tasks.

Together, the DSM model and method fulfill the requirements of modularity measurement. However, many steps remain for a full reconfigurability measurement process. In particular, little has been mentioned with regard to the sequence of functions or their allocation to subsystems. This gap may be fulfilled by newly developed concepts such as production degrees of freedom [15][16]. Nevertheless, the DSM appears as a promising tool for the ultimate goal of a comprehensive reconfigurability measurement process.

REFERENCES

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

[2] L. Bongaerts. *Integration of Scheduling and Control in Holonic Manufacturing Systems*. PhD thesis, Department Werktuigkunde Afdeling Productietechnieken. Katholieke Universiteit Leuven, 1998.

[3] L. Bongaerts, L. Monostori, D. McFarlane, and K. Botond. Hierarchy in distributed shop floor control. *Computers in Industry*, 43(2):123–137, 2000.

[4] C.G. Cassandras. Autonomous material handling in computer integrated manufacturing. In A. Kussiak, editor, *Modelling and Design of Flexible Manufacturing Systems*. Elsevier Science Publishers, Amsterdam, 1986.

[5] R.H. Cerni and L.E. Foster. *Instrumentation for Engineering Measurment*. John Wiley and Sons, New York, 1962.

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

[7] E.W. Dijkstra. Beyond structured programming: An introduction to the principles of software metrics. *Journal of Structured Programming*, 11(1), 1990.

[8] D.M. Dilts, N.P. Boyd, and H.H. Whorms. Evolution of control architectures for automated manufacturing systems,. *Journal of Manufacturing Systems*, 10(1):79–93, 1991.

[9] N. Duffie, R. Chitturi, and J. Mou. Fault tolerant heterarchical control of heterogeneous manufacturing entities. *Journal of Manufacturing Systems*, 7(4):315–328, 1988.

[10] N. Duffie and R. Piper. Non hierarchical control of a flexible manufacturing cell. *Robotics and Computer Integrated Manufacturing*, 3(2):175–179, 1987.

[11] L.O. Ejiogu. *Software Engineering with Formal Metrics*. QED Technical Publishing Group: McGraw-Hill Book Company, Boston, 1991.

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

[13] A.M. Farid. *Reconfigurability Measurement in Automated Manufacturing Systems*. PhD thesis, University of Cambridge: Institute for Manufacturing, 2007.

[14] 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 *submitted to: IEEE 2006 Workshop on Distributed Intelligent Systems Collective Intelligence and its Applications*, Prague, Czech Republic, 2006.

[15] A.M. Farid and D.C. McFarlane. A development of degrees of freedom for manufacturing systems. In *IMS'2006: 5th International Symposium on Intelligent Manufacturing Systems: Agents and Virtual Worlds*, Sakarya, Turkey, 2006.

[16] A.M. Farid and D.C. 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.

[17] N. Gayed, D. Jarvis, and J. Jarvis. A strategy for the migration of existing manufacturing systems to holonic systems. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 319–324, San Diego, 1998.

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

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

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

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

[22] C.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.

[23] P. Kidd. *Agile Manufacturing: Forging new Frontiers*. Addison-Wesley, Wokingham, 1994.

[24] 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.

[25] S.C. Kumar and M.P. Chandrasekharan. Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrics in group technology. *International Journal of Production Research*, 28(2):233–243, 1990.

[26] P. Leitao. *An Agile and Adaptive Holonic Architecture for Manufacturing Control*. PhD thesis, University of Porto: Faculty of Engineering, 2004.

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

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

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

[30] M.G. Mehrabi, A.G. Ulsoy, and Y. Koren. Reconfigurable manufacturing systems: key to future manufacturing. *Journal of Intelligent Manufacturing*, 11(4):403–419, 2000.

[31] T.U. Pimmler and S.D. 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.

[32] J.B. Pine. *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Pressed, Boston, MA, 1993.

[33] K. Preiss, S.L. Goldman, and R.N. Nagel. *Cooperate to Compete: Building Agile Business Relationships*. Van Nostrand Reinhold, London, 1996.

[34] R.M. Setchi and N. Lagos. Reconfigurability and reconfigurable manufacturing systems – state of the art review. In *Innovative Production Machines and Systems: Production Automation and Control State of the Art Review*, chapter 5, pages 131–143. Innovative Production Machines and Systems: Network of Excellence, Cardiff, United Kingdom, 2005.

[35] M.E. Sosa, S.D. Eppinger, and C.M. 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.

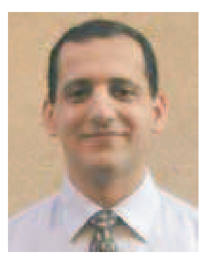
[36] S. Sosale, M. Hashemian, and P. Gu. Product modularization for reuse and recycling. In *American Society of Mechanical Engineers, Design Engineering Division (Publication) DE*, volume 94, pages 195–206, Dallas, TX, USA, 1997.

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

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

[39] J. Wyns. *Reference Architecture For Holonic Manufacturing Systems - the key to support to evolution and reconfiguration*. PhD thesis, Department Werktuigkunde Afdeling Productietechnieken. , Katholieke Universiteit Leuven, 1999.

[40] Y. Zhang, J.K. 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.



Amro M. Farid received his Sc.B. from the MIT Mechanical Engineering Department where he focused on system dynamics and control. He also completed his Sc.M within the same department in the field of Quantum Information Processing (QIP). Amro is currently pursuing a PhD at the Insitute for Manufacturing on the design and evaluation of Reconfigurable Manufacturing Systems.

Duncan C. McFarlane is Professor of Service and Support Engineering at the Cambridge University Engineering Department, and head of the Distributed Information & Automation Laboratory within the Institute for Manufacturing. He is also Director of the Cambridge Auto-ID Lab and Research Director of two industrially supported activities: the Service and Support Engineering Programme and the Aero ID Programme. He has been involved in the design and operation of automation and information system for the manufacturing supply chain for twenty years and his research focuses on distributed industrial automation, reconfigurable systems, RFID integration and valuing industrial information.