ELSEVIER

Computers in Industry 33 (1997) 261-269

COMPUTERS IN INDUSTRY

A framework for classifying flexibility types in manufacturing

John P. Shewchuk [a,*], Colin L. Moodie [b]

[a] Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0118, USA
[b] School of Industrial Engineering, Purdue University, West Lafayette, IN 47907-1287, USA

Abstract

In today's dynamic and uncertain manufacturing environment, *flexibility* is one of the most sought-after properties for manufacturing systems. Despite this interest, flexibility remains poorly understood in theory and utilised in practice. One of the underlying reasons for this is the fact that there is no general agreement on how to define flexibility: over 70 terms can be found in the literature. This paper is concerned with developing a unified framework for classifying the various types of flexibilities found in manufacturing. The framework is an attempt to provide both a mechanism for classifying existing flexibility types and understanding their relationships, and a foundation for future efforts in defining flexibility types and measures. The framework is developed based upon the architectural concepts of a 'basic mechanism model' and system design activity, and a system/environment model. Examples of how some existing flexibility types may be classified according to the framework are given. © 1997 Elsevier Science B.V.

Keywords: Flexibility; Manufacturing systems; Modelling; Architectures

1. Introduction

A great deal of research on defining various types of flexibilities in manufacturing has occurred over the last two decades. Despite this, there is no general agreement on how to define flexibility. At the outset, this is due to the multi-dimensional nature of flexibility and the various views of flexibility which result: flexibility has been viewed and studied as a physical property, an attribute of decision making, an economic indicator, and a strategic tool. In a comprehensive survey of the literature, Sethi and Sethi [1] reported that at least 50 terms exist for the various types of flexibilities studied. Furthermore, they found that several terms refer to the same flexibility type in many cases, and that definitions for flexibility types are often imprecise and conflicting, even for identical terms.

This problem results from the fact that researchers have by-and-large defined flexibility types based upon a limited view of the manufacturing system, reflecting their own particular areas of interest and biases. The flexibility types which they define are consequently based upon a wide range of models and assumptions, many of which are not always stated.

To remedy the situation, a unified framework for classifying flexibility types is required. Using such a framework, flexibility types can be defined in a common manner, even if they have been derived based upon different models and assumptions. Such a framework will provide not only a mechanism for classifying existing flexibility types, but also a foundation for future efforts in defining flexibility types and measures.

* Corresponding author.

PII S0166-3615(97)00032-8

2. Background

Despite the large amount of effort expended in defining flexibility types, little work has been done to develop a framework for classifying such types. Researchers have concentrated simply on identifying and defining flexibility types, based upon specified models and assumptions. Several researchers, however, have noted and described some of the basic attributes of flexibility. Mandelbaum [2] and Slack [3] define basic categories of flexibility. Relative vs. absolute views of flexibility are noted by Gupta and Buzacott [4] and Chryssolouris and Lee [5]. Finally, the time-dependent nature of flexibility has been suggested and discussed by various researchers, including Gustavsson [6] and Gupta and Buzacott [4]. These works shall be elaborated on in Section 4.

3. Modelling framework

The modelling framework for developing a framework for classifying flexibility types consists of three items: a 'basic mechanism model', a model of the manufacturing system design activity, and a system/environment model. Each of these items is now described.

3.1. Basic mechanism model

The basic mechanism model is shown in Fig. 1. First, there is a physical *mechanism* which is capable of transforming certain inputs into certain outputs. Such transformations may be either physical (i.e. involving matter and energy) or informational in nature. Next, there is a set of *objectives* (desired outputs and any additional performance-related criteria) and associated inputs to the mechanism. The mechanism attempts to transform the given inputs into the desired outputs, while satisfying any additional performance-related criteria.

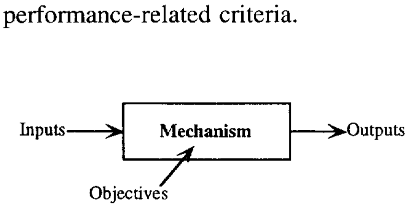

Fig. 1. Basic mechanism model.

If the transformation is *possible* (i.e. desired outputs can be attained over an instantaneous time interval), the mechanism is said to be *capable* for the transformation. If the transformation can be *maintained* over an indefinite period of time, i.e. 'steady-state' is possible, the mechanism is said to be *stable* for the transformation. The exact meaning of steady-state, and of course how to determine if steady state has been reached, must be defined. Note also that capable and stable say nothing about performance.

In order for a mechanism to be capable and stable for a given set of objectives and associated inputs, it may need to be *configured.* This consists of changing the basic structure and/or functionality of the mechanism.

This basic model is applicable to mechanisms at all levels of decomposition. That is, the mechanism may be an individual machine at the lowest level, or the entire manufacturing system, seen as a 'black box', at the top level. For an individual machine, mechanism configuration may consist of setting up the machine, whereas for the entire manufacturing system it may consist of relocating equipment, changing the scheduling approach and/or algorithms, etc.

3.2. Manufacturing system design activity

In designing a manufacturing system, the set of all activities which must be executed by the system, including all information regarding timing, precedence, connectivity, etc. must be specified. This set of *functional requirements* can be specified independent of any physical mechanisms which actually implement these activities [7]. Once the functional requirements have been specified, a great variety of manufacturing system designs, in terms of both component selection and system configuration, are possible. We shall refer to this set as the *manufacturing system solution space*. The actual manufacturing system design to be used must then come from this solution space.

In actuality, there will be many constraints on selection of the manufacturing system design to use. For simplicity, the sole constraint considered here is the capital available for the initial purchase, installation, and commissioning of the manufacturing system.

3.3. System / environment model

The interaction of the manufacturing system with its environment is defined by the inputs and outputs to the system. Though there are of course a great many possible inputs and outputs, we consider the single interaction to be that of the stream of products flowing through the system. Thus, the manufacturing environment can be defined in terms of two items: product requirements and production requirements. The variables defining the manufacturing environment, and their relationships, are shown in Fig. 2 and described below.

3.3.1. Product requirements

Any product which is possible for the given manufacturing environment can be specified in terms of its *processing data*. This data is similar to the process plan in that it describes what is required to produce the product, but it differs in that it is independent of any particular manufacturing system. Thus, process routings replace machine routings, work/operation replaces operation run time, etc. The set of variables used to define processing data for any product are referred to as the *processing data variables*.

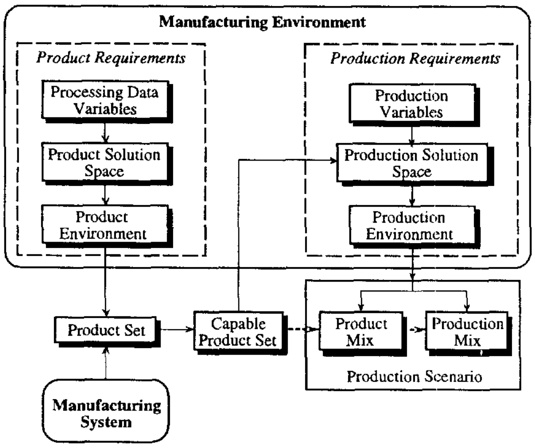

Fig. 2. Variables defining the manufacturing environment and their relationships.

The set of products which are *possible* for the given manufacturing environment and set of processing data variables forms the *product solution space*. The product solution space is defined by the *product solution space specification*, a description of (i) the set of values defining bounds on (or solution space of) each processing data variable, and (ii) any constraints on processing data combinations (i.e. products). All products found in the product solution space can be generated from the product solution space specification.

While the product solution space specifies which products are possible, it does not specify which types of products are *likely*. For example, the degree of product similarity, or the extent to which alternate processes can be used for operations, may be low in one manufacturing environment, high in the next. Thus, it is important to capture this information, so that flexibility can be analysed relative to the product configurations which are likely to occur. The set of products which are considered likely for a given manufacturing environment, set of processing data variables, and product solution space is referred to as the *product environment*. The product environment is defined by the *product environment specification*, a description of (i) random variables used for generating processing data variable values, and (ii) expressions indicating the manner in which each product may differ from its predecessor (e.g. quantity of identical operations).

3.3.2. Production requirements

Given a manufacturing system and set of products, the system will be capable of processing a certain subset of these products. We refer to such a subset as a *capable product set*. There are many possible scenarios for manufacturing products of a given capable product set: such scenarios can be defined in terms of two variables:

1. The subset of products in the capable product set to be produced simultaneously. We shall refer to any such subset as a *product mix*.
2. Values for the various *production variables* (batch sizes, batch compositions, arrival rates, etc.) for the given product mix. We shall refer to any combination of such values as a *production mix*.

Together, a given product mix and production mix define a *production scenario*. For a given capable product set and set of production variables, the set of all production scenarios which are possible is referred to as the *production solution space*. The production solution space is defined by the *production solution space specification*, a description of (i) any constraints on product mix combinations, (ii) bounds on production variables, and (iii) any constraints on production variable value combinations. All production scenarios found in the production solution space can be generated from the production solution space specification.

In similar manner to the product solution space, the production solution space defines which production scenarios are possible, but not which are *likely*. One company may produce the same products continually (constant product mix), but have to cope with changing production rates; another may have relatively steady production rates but varying batch sizes, etc. Thus, when analysing the flexibility of a manufacturing system, the production scenarios considered should reflect the true nature of production, or *production environment*. The production environment is defined by the *production environment specification*, a description of the random variables used to generate product mixes and production mixes.

4. Framework for classifying flexibility types

Based upon the above modelling framework, we can develop a framework for classifying flexibility types. The framework consists of three axis: primitive flexibility type, level of requirements specification, and time frame. These are defined as follows.

4.1. Primitive flexibility type

Analysis of the basic mechanism model leads to the identification of three different types of flexibility as follows:

1. Given a mechanism and set of possible sets of objectives, the first question is whether or not it is possible for the mechanism to satisfy each set of objectives (i.e. be both capable and stable), assuming suitable inputs are provided. It is intuitive that the larger the subset of such objective sets that the mechanism is capable of satisfying, the more flexible the mechanism is. We refer to this as *range flexibility* after Slack [3], who defines range flexibility as the 'total envelope of capability or range of states which the production system or resource is capable of attaining'.
2. Consider again that the mechanism may be faced with a set of possible sets of objectives, but now assume that one of these sets of objectives must be selected (equivalently, we can consider a single set of objectives and a variety of mechanisms available). It is intuitive that the easier it is for the mechanism to satisfy a given set of objectives, the more flexible the situation is with regards to selecting those objectives. The term 'ease' must of course be defined: this could refer to cost, energy consumption, time required, etc. We shall refer to this as *action* flexibility, as it is concerned with taking a particular course of action. Note that this is different from Mandelbaum's [2] definition of action flexibility as the 'capacity to take new action to meet new circumstances'.

If the objective is to configure the mechanism for a subsequent set of objectives, action flexibility is equivalent to Slack's [3] definition of *response flexibility* as the 'ease (in terms of cost, time, or both) with which changes can be made within the capability envelope'.

3. Finally, consider that we are given a mechanism, set of objectives, and inputs, where the mechanism is capable of satisfying the set of objectives. Now consider that changes occur to either the mechanism and/or inputs. Such changes will likely have some effect on the mechanism output. It is intuitive that the less the output is affected by such changes, the greater the ability of the mechanism to cope with change and thus the more flexible the mechanism. We refer to this as *state* flexibility, after Mandelbaum [2], who defines state flexibility as the 'capacity to continue functioning effectively despite change'. Note that state flexibility can be measured relative to one of three possible variable types: capability, stability, and performance.

It is postulated that each of the many flexibility types identified in the literature can in fact be shown to be one of (or based upon a combination of) these three types. Due to this fundamental nature, these shall be referred to as *primitive* flexibility types: various flexibility types can then be established based upon each primitive.

4.2. Level of requirements specification

Level of requirements specification refers to which variables, of those required to specify a production scenario, are fixed and which are free. A different set of flexibility measures can be developed at each such level. The requirements specification level attribute of flexibility types has not been previously identified or defined in the literature. Various researchers have noted some of the characteristics of this attribute, however. Chryssolouris and Lee [5], for example, state that there are two views of flexibility: the first where flexibility is an 'intrinsic attribute' of a manufacturing system, and the second where flexibility is a 'relative attribute', depending upon the external demands placed upon the system. Other researchers noting this characterisation include Gupta and Buzacott [4].

For the system/environment model of Section 3.3, five levels of requirements specification can be identified: these levels are described below. To aid in understanding the nature of the variables which are fixed at each level, an example of such variables is presented in each case.

Level I: At Level I, the functional requirements for the manufacturing system, the manufacturing system solution space, and the capital constraint for the system design are all given. Level I measures then indicate how flexible the system is compared to what it *could* be for the given constraints. Measures which indicate that one system (or element thereof) is more flexible than another, when the comparison is made without reference to any particular products or production requirements (e.g. FMS vs. transfer line), are Level I measures.

An example of functional requirements for a manufacturing system are the required part-producing capabilities the system must possess. We consider the case where two types of capabilities are considered: setup type (e.g. fixturing requirements) capabilities and process capabilities. Functional requirements can then be specified in terms of three items: (i) the set of setup types the system must be capable of performing, denoted by S^*, (ii) the set of processes which the system must be capable of performing, P^*, and (iii) relationships indicating which processes must be able to be performed with which setup types. We can describe these functional requirements in terms of a P/S (process/setup type) *matrix*. Fig. 3 shows two different P/S matrices for the case where $S^* = 1,2,3,4$ and $P^* = 1,2,3,4,5,6$.

P \ S	1	2	3	4
1	x			
2	x	x	x	
3	x	x	x	
4	x	x	x	
5				
6				

a) P/S matrix #1

P \ S	1	2	3	4
1	x	x	x	
2	x	x	x	x
3	x	x	x	x
4	x	x	x	x
5	x	x	x	x
6	x	x	x	x

b) P/S matrix #2

Fig. 3. Examples of functional requirements of manufacturing systems (P/S matrices).

The second item given is the manufacturing system solution space For simplicity, we consider that the manufacturing system is described in terms of processors (e.g. machines) only. The variables used for specifying the manufacturing system, and bounds (solution space) specification for each variable, are as follows:

- N_{PRT} Quantity of processor types, $LB(N_{PRT}) \leq N_{PRT} \leq UB(N_{PRT})$
- Ns_i Quantity of setup types at processor type i, $1 \leq Ns_i \leq UB(Ns)$, $UB(Ns)$ = maximum quantity of setup types possible at any processor type
- $\boldsymbol{S}_i$ Set of setup types at processor type i, $\{s_{i1}, s_{i2}, \ldots, s_{iN}\}, s_{ij} \in \boldsymbol{S}^*$, $s_{ij} \notin \{s_{i1}, s_{i2}, \ldots, s_{ij-1}\}$, $j = 1, \ldots, N, N = Ns_i$
- Np_i Quantity of processes at processor type i, $1 \leq Np_i \leq UB(Np)$, $UB(Np)$ = maximum quantity of processes possible at any processor type
- $\boldsymbol{P}_i$ Set of processes at processor type i, $\{p_{i1}, p_{i2}, \ldots, p_{iN}\}, p_{ij} \in \boldsymbol{P}^*$, $p_{ij} \notin \{p_{i1}, p_{i2}, \ldots, p_{ij-1}\}$, $j = 1, \ldots, N, N = Np_i$
- M_i Quantity of processors of type i, $M_i^3 \geq 1$

Values for the bounding variables $UB(Ns)$ and $UB(Np)$ depend upon the current level of (machine) technology and marketing factors (i.e. what vendors are offering). We shall let $UB(Ns) = 3$ and $UB(Np) = 4$ for this example. Values for the bounding variables $LB(N_{PRT})$ and $UB(N_{PRT})$ are dependent upon the P/S matrix, $UB(Ns)$, and $UB(Np)$: an algorithm is therefore required to calculate these values dynamically. Once bounding variable values have been specified, the manufacturing system solution space depends upon any constraints on possible combinations of manufacturing system variable values. We assume no such constraints for this example.

The final item specified is the capital constraint, for example, available capital = \$300 K.

Level II: At Level II, the manufacturing system and the product solution space are given. Level II measures indicate the ability of the system to process the universe of products which are deemed possible for the manufacturing environment.

Four different manufacturing systems, two for each set of functional requirements (Fig. 3), the above manufacturing system solution space, and the above capital constraint, are illustrated in Fig. 4. In order to determine if the capital constraint is met for a given manufacturing system, a cost model and cost data are required to determine the cost of a given system, C_{SYS}. A simple cost model based upon processors only is used here: the cost of processor type $i = c_B + Ns_i * c_S + Np_i * c_P$, where c_B is the base cost of the processor, c_S is the cost of providing each setup type, and c_P is the cost of providing each process. The cost data used for the example systems is $c_B = \$20$ K, $c_S = \$2.5$ K, and $c_P = \$5$ K.

The second item given is the product solution space. We assume the following processing data variables and bounds specifications:

N_{OPS_m} Quantity of operations for product type m, $LB(N_{OPS}) \le N_{OPS_m} \le UB(N_{OPS})$

s_{mn} Setup type for nth operation of product type m, $s_{mn} \in S^*$, $n = 1,\ldots,N_{OPS_m}$

p_{mn} Process for nth operation of product type m, $p_{mn} \in P^*$, $n = 1,\ldots,N_{OPS_m}$

v_{mn} Setup time for nth operation of product type m, $v_{mn} \in V$, V = set of possible values

w_{mn} Processing time for nth operation of product type m, $w_{mn} \in W$, W = set of possible values

The product solution space specification then consists of (i) values for the bounding variables $LB(N_{OPS})$, $UB(N_{OPS})$, S^*, P^*, V, W, and (ii) any constraints on processing data combinations. For example, the specification could consist of the values $LB(N_{OPS}) = 3$, $UB(N_{OPS}) = 8$, $V = \{5.0 \text{ (minutes)}, 9.0, 16.0\}$, $W = \{20.0 \text{ (minutes)}, 22.0, 24.0, 26.0, 30.0, 40.0\}$, and S^* and P^* as previously defined, and the constraints that the first operation must utilise either process 1 or 2 and that process 5 must follow process 3. Algorithms are required to determine the size of the product solution space and/or generate the corresponding products for the given bounding variables and any set of constraints on processing data combinations.

Level III: The requirements specification variables given at Level III are the manufacturing system, product solution space, and product environment. Level III measures indicate the ability of the system to process those products which are planned, anticipated, or likely to be encountered in the manufacturing environment.

An example product environment for the previously defined product solution space is:

1. The following random variables for generating processing data variable values:

N_{OPS_m} $U_D(LB(N_{OPS}),UB(N_{OPS}))$

s_{mn} $p(1) = 0.20$, $p(2) = 0.50$, $p(3) = 0.30$

p_{mn} $p(1) = p(6) = 0.10$, $p(2) = p(5) = 1.15$, $p(3) = p(4) = 0.25$

v_{mn} $U_S(V)$

w_{mn} $U_S(W)$

where $U_D(a,b)$ indicates a discrete uniform distribution over the interval $[a,b]$, $U_S(S)$ indicates a discrete uniform distribution over all items in set S, and $p(a) = P\{X = a\}$ for discrete random variable X.

2. The constraint that, at most, 30% of the operations for any product type $m > 1$ may differ from those of product type $m - 1$. Furthermore, both the quantity and selection of operations which remain constant between product types are random.

An algorithm is then required to generate products according to the specified random variables such that the specified constraint is met.

Level IV: At Level IV, the manufacturing system, product solution space, product environment, and production solution space are all given. Level IV measures indicate the ability of the system to handle

$N_{PRT} = 5$

S_i:	1	1	2	3	1,2,3
P_i:	1	2,3	2,3	2,3	4
M_i:	1	3	2	2	1

a) P/S matrix #1, manufacturing system #1 (C_{SYS} = \$287.5K)

$N_{PRT} = 9$

S_i:	1	1	1	2	2	3	3	1,2	3
P_i:	1	2	3	2	3	2	3	4	4
M_i:	1	1	1	1	2	1	1	1	1

b) P/S matrix #1, manufacturing system #2 (C_{SYS} = \$277.5K)

$N_{PRT} = 5$

S_i:	1,2,3	1,2,3	4	4	4
P_i:	1,2,3,4	5,6	2,3	4,5	6
M_i:	1	3	1	2	1

c) P/S matrix #2, manufacturing system #1 (C_{SYS} = \$285K)

$N_{PRT} = 8$

S_i:	1	2	3	1,2,3	1,2,3	4	4	4
P_i:	1,2	1,2	1,2	3,4	5,6	2,3	4,5	6
M_i:	1	1	1	1	1	1	1	2

d) P/S matrix #2, manufacturing system #2 (C_{SYS} = \$292.5K)

Fig. 4. Examples of manufacturing systems for given functional requirements and capital constraints.

all possible production scenarios for any set of products which is planned, anticipated, or likely to be encountered in the manufacturing environment.

An example production solution space is the case of batch production, where a single input stream exists and all batches are identically composed. In this case, the product mix is equivalent to the set of different product types found in each batch. The production variables describing this case, and their bounds specifications, are as follows:

T_j Time between batch arrivals j and $j-1$, $T_j \in \boldsymbol{T}$, $\boldsymbol{T}$ = set of possible values

Q_B Total quantity of products in each batch, $Q_B \in \{\text{LB}(Q_B), \text{LB}(Q_B) + ss_q, \ldots, \text{LB}(Q_B) + n_q * ss_q\}$

q_k Quantity of kth product type in each batch, $1 \le q_k \le Q_B$

Let $\boldsymbol{Pc}_i$ denote a capable product set for manufacturing system i, and $\boldsymbol{P}_m$ a product mix (i.e. some subset of $\boldsymbol{Pc}_i$). The product solution space specification then consists of (i) constraints on possible product mixes (i.e. $\boldsymbol{P}_m$), (ii) values for the bounding variables $\boldsymbol{T}$, LB(Q_B), n_q, and ss_q, and (iii) constraints on production variable value combinations. For example, this specification could consist of the constraint that $\|\boldsymbol{P}_m\| \ge [\|\boldsymbol{Pc}_i\|/2]^+$, the values $\boldsymbol{T} = \{7.0$ (hours), 8.0, 10.0$\}$, LB(Q_B) = 80 items, $n_q = 4$, and $ss_q = 10$ items, and the constraint that one product type shall dominate each batch and all other product types are equally represented. This last constraint can be stated as

$$
\begin{aligned}
q_k &= Q_B & \|\boldsymbol{P}_m\| = 1 \\
&= \left[Q_B/(\|\boldsymbol{P}_m\| + a)\right]^- & k \ne k^*, \|\boldsymbol{P}_m\| > 1 \\
&= Q_B - (\|\boldsymbol{P}_m\| - 1) & k = k^*, \|\boldsymbol{P}_m\| > 1 \\
&\quad * \left[Q_B/(\|\boldsymbol{P}_m\| + a)\right]^- &
\end{aligned}
$$

where k^* indicates the dominant product type, which will dominate all other product types by as close to as possible, but not less than, the ratio $(a+1)$:1. For example, if $Q_B = 100$, $\|P_m\| = 6$, $a = 2$, and $k^* = 3$, the above gives $q_1 = q_2 = q_4 = q_5 = q_6 = 12$, $q_3 = 40$.

Level V: The requirements specification variables given at Level V are the manufacturing system, product solution space, product environment, production solution space, and production environment. Level V measures indicate the ability of the system to handle those production scenarios which are planned, anticipated, or likely to be encountered for any set of products which is planned, anticipated, or likely to be encountered in the manufacturing environment.

An example production environment for the previously defined production solution space consists of the following random variables for generating product mixes and production mixes (let P_k = kth product type in $\boldsymbol{P}_m$):

$\|\boldsymbol{P}_m\|$ $U_D([\|Pc_i\|/2]^+, \|\boldsymbol{Pc}_i\|)$

P_k $U_S(\boldsymbol{Pc}_i)$ such that $P_k \notin \{P_1, P_2, \ldots, P_{k-1}\}$

T_j $p(7.0) = 0.20$, $p(8.0) = 0.40$, $p(10.0) = 0.40$

Q_B $p(80) = p(120) = 0.10$, $p(90) = p(110) = 0.20$, $p(100) = 0.40$

q_k $k^* = U_D(1, \|\boldsymbol{P}_m\|)$

4.3. Time frame

Time frame refers to the time interval over which the flexibility measure is calculated. The basic differentiation is between static measures (those dependent only upon system structure, i.e. upon an instanta-

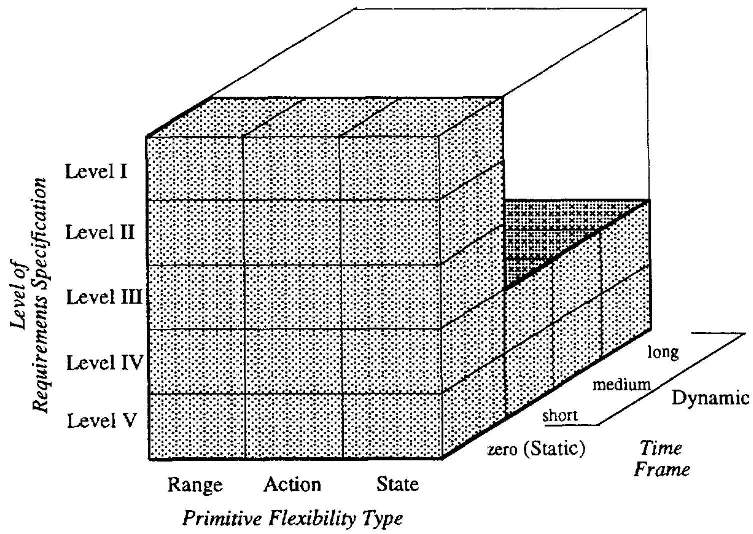

Fig. 5. Framework for classifying flexibility types.

neous time interval or 'snapshot' of the system) and dynamic measures (those dependent upon system behaviour, i.e. based upon system response over a given non-zero, finite-length time interval). Dynamic measures can be further divided in some manner based upon interval length. A generic method for this, which is used here, is to simply select three time intervals: short, medium, and long. Suitable values for each time interval may then be selected as required. This particular set of time intervals has been suggested by various researchers, e.g. Gupta and Buzacott [4] and Gustavsson [6]. The latter also suggest the following values for these time intervals: several minutes to several hours (short), several days to several months (medium), and several months to several years (long). It should be noted that a more formal approach to defining dynamic measure time intervals and interval lengths would be to derive these items from the planning and control architecture upon which the system is based. This remains for future research.

We note that time frame and level of requirements specification are not independent. No dynamic data is present at Levels I, II and III: consequently, static measures only may exist at these levels. Both static and dynamic measures may exist, however, at Levels IV and V.

4.4. Resulting framework

The resulting modelling framework for classifying flexibility types is shown in Fig. 5. At each three-dimensional shaded 'section' of the framework, flexibility types having common attributes can be located.

5. Examples

Examples of flexibility measures found at each level of requirements specification are as follows:

Level I—Routing flexibility [8], machine flexibility [1,8], expansion flexibility [1,9].

Level II—Production flexibility [1,9], process flexibility [1].

Level III—Product flexibility [9], job flexibility [10].

Level IV—Volume flexibility [9].

Level V—Market flexibility [1], demand flexibility [11].

6. Conclusions

A framework for classifying the various types of flexibilities found in manufacturing has been proposed. The framework was based upon various architectural concepts and a system/environment model. It is important to note that the framework developed is *generic* in that neither variables defining the manufacturing environment (e.g. processing data variables), nor time frame intervals, were specified. A variety of *particular* frameworks are thus possible, depending upon the variables selected. For example, considering the processing data variables, one framework could be for single-level BOM (bill-of-materials) items with single-resource routings, whereas another could be for multi-level BOM items with multiple-resource routings.

The utility of this framework is that it (i) allows us to classify flexibility types in order to see which types share common attributes and how such types are related to one another, (ii) assists us in identifying which of the many terms already proposed by various researchers in fact refer to the same flexibility type, and (iii) provides a mechanism for graphically representing how various flexibility types are related to one another and thus an aid for visualising and understanding these relationships. One thing the framework cannot be used for, however, is for *identifying* all of the flexibility types found in manufacturing. In order to do this in a structured manner, one must start with a suitable system/environment model (such as a particularised instance of that described here) and manufacturing system architecture, and then *derive* the various flexibility types based upon the elements comprising these models and their relationships with one another. The framework is nonetheless of value for this pursuit in that it helps us to determine what we are looking for. Once the various flexibility types have been found in this manner, they can then be mapped to the framework for the aforementioned reasons.

Continued research is first and foremost required to prove the utility of the proposed framework by mapping existing flexibility types to it. This will not necessarily be possible in all cases, as the intended level of requirements specification is not always clear. Efforts must then be focused on deriving flexibility types from a specified architecture and system/environment model as described above. It is hoped that the framework developed here will prove to be of value in performing this difficult task.

References

[1] A.K. Sethi, S.P. Sethi, Flexibility in manufacturing: a survey, International Journal of Flexible Manufacturing Systems 2 (1990) 289–328.

[2] M. Mandelbaum, Flexibility in decision making: an exploration and unification, Ph.D. Thesis, Department of Industrial Engineering, The University of Toronto, Toronto, Canada, 1978.

[3] N. Slack, The flexibility of manufacturing systems, International Journal of Operations and Production Management 7 (4) (1987) 35–45.

[4] D. Gupta, J.A. Buzacott, A framework for understanding flexibility of manufacturing systems, Journal of Manufacturing Systems 8 (2) (1989) 89–97.

[5] G. Chryssolouris, M. Lee, An assessment of flexibility in manufacturing systems, Manufacturing Review 5 (2) (1992) 105–116.

[6] S. Gustavsson, Flexibility and productivity in complex production processes, International Journal of Production Research 22 (5) (1984) 801–808.

[7] T.J. Williams, The Purdue Enterprise Reference Architecture, Instrument Society of America, Research Triangle Park, NC, 1992.

[8] M.F. Carter, Designing flexibility into automated manufacturing systems, in: K.E. Stecke, R. Suri (Eds.), Proceedings of the 2nd ORSA/TIMS Conference on Flexible Manufacturing Systems, Elsevier, Amsterdam, 1986, pp. 107–118.

[9] J. Browne, D. Dubois, K. Rathmill, S.P. Sethi, K.E. Stecke, Classification of flexible manufacturing systems, The FMS Magazine, April 1984.

[10] J.A. Buzacott, The fundamental principles of flexibility in manufacturing systems, Proceedings of the 1st International Conference in Flexible Manufacturing Systems, North-Holland, Amsterdam, 1982, pp. 23–30.

[11] Y.K. Son, C.S. Park, Economic measure of productivity, quality and flexibility in advanced manufacturing systems, Journal of Manufacturing Systems 6 (3) (1987) 193–207.