
Article:

Koh, S.C.L., Gunasekaran, A. and Saad, S.M. (2006) Parts verification for multi-level-dependent demand manufacturing systems: a recognition and classification structure. The International Journal of Advanced Manufacturing Technology, 31 (3-4). pp. 305-315. ISSN 1433-3015

https://doi.org/10.1007/s00170-005-0184-9

Published paper

Koh, S.C.L., Gunasekaran, A. and Saad, S.M. (2006) *Parts verification for multi-level-dependent demand manufacturing systems: a recognition and classification structure.* The International Journal of Advanced Manufacturing Technology, 31 (3-4). pp. 305-315.

Parts verification for multi-level dependent demand manufacturing systems: A recognition and classification structure

S.C.L. Koh[1], S.M. Saad[2], A. Gunasekaran[3]

[1] University of Sheffield, Management School, 9, Mappin Street, Sheffield, S1 4DT, UK

Tel: +44 (0) 114 222 3395 Fax: +44 (0) 114 222 3348 Email: S.C.L.Koh@sheffield.ac.uk

[2] Sheffield Hallam University, School of Engineering, City Campus, Sheffield, S1 1WB, UK

Tel: +44 (0) 114 225 3393 Fax: +44 (0) 114 225 3433 Email: S.Saad@shu.ac.uk

[3] University of Massachusetts, Management Department, 285 Old Westport Road, North Dartmouth, MA 02747-2300, USA;

Tel: +1 (508) 999 9187, Fax: +1 (508) 999-8776, Email: Agunasekaran@umassd.edu

Keywords: Parts verification, manufacturing systems, MRP

Abstract

This research has developed and implemented a part recognition and classification structure to execute parts verification in a multi-level dependent demand manufacturing system. The part recognition algorithm enables the parent and child relationship between parts to be recognised in a finite-capacitated manufacturing system. This algorithm was developed using SIMAN simulation language and implemented in a multi-level dependent demand manufacturing simulation model. The part classification structure enables the modelling of a multi-level dependent demand manufacturing between parts to be carried out effectively. The part classification structure was programmed using Visual Basic Application (VBA) and was integrated to the work-to-list generated from a simulated MRP model. This part classification structure was then implemented in the multi-level dependent demand manufacturing simulation model. Two stages of implementation, namely parameterisation and execution, of the part recognition and classification structure were carried out. A real case study was used and five detail steps of execution were processed. Simulation experiments and MRP were run to verify and validate the part recognition and classification structure. The results led to the conclusion that implementation of the recognition and classification structure has effectively verified the correct parts and sub-assemblies used for the correct product and order. No parts and sub-assemblies shortages were found, and the quantity required was produced. The scheduled release for some orders was delayed due to overload of the required resources. When the loading is normal, all scheduled release timing is adhered to. The recognition and classification structure has a robust design; hence it can be easily adapted to new systems parameter to study a different or more complex case.

1. Introduction

In a multi-level dependent demand manufacturing system, order execution is controlled by a series of work-to-list. A work-to-list contains the job sequence for producing every demanded parts and sub-assemblies for each product. Such work-to-list can be obtained from a production planning and scheduling system. Much research has been carried out in modelling a multi-level dependent demand manufacturing system. However, little effort is emphasised on parts verification to ensure that the correct parts are being produced and allocated to its specified demand. This aspect is important to enhance customer service level by ensuring that their order will not be compensated with other orders. Hence, to effectively model order execution for the correct part, the correct quantity and timing becomes increasingly essential.

In this research, we model Material Requirements Planning (MRP) as our production planning and scheduling system. MRP was designed to ideally operate within stable and predictable batch manufacturing environments. It is a multi-level dependent demand system that is logically linked by a set of backward scheduling techniques. Multi-level Bill of Material (BOM) governs parts demand dependency. The backward scheduling starts from the Master Production Schedule (MPS), which defines the product required, its ordered quantity and its due date. It then explodes the BOM to identify all parts required. This gives the gross requirements. Inventory data and schedule receipts are then offset to compute the net requirements. To determine the release date, due date of the order is offset with the planned lead-time. This netting process starts from the highest level of the BOM, working to the lowest level. This backward scheduling is also referred as MRP run. The output from this run is a Planned Order Release (POR) schedule, which is equivalent to a work-to-list. This work-to-list typically contains customer order number, part number, routing number, net requirement, release date and due date for all parts in the MPS. The fundamental function of the work-to-list is that it establishes that parts at lower levels of the BOM in the order book must be completed before parts at the related higher levels begin their operations.

MRP assumes infinite capacity when generating the POR schedule. This implies the schedule is over-optimistic because resources required are expected to be available in the shop floor. Unfortunately, this is not the case in a realistic manufacturing environment. For example, machine breakdown could distort the schedule thus results in delay in the current and the subsequent operations. Additionally, MRP also assumes fixed lead-time, which relaxes variation in lead-times during production. For example, suppliers' tardiness is not reflected during planning. These assumptions often cause deviations between the POR schedule and the finite-capacitated manufacturing environments, particularly under the condition of uncertainty where disruption to production is inevitable.

Extensive research has been carried out to study the performance of a finite-capacitated manufacturing environment under MRP planning and scheduling rules, and simulation modelling was found to be the most common method used (Koh *et al*, 2002). However, little research looked at the issue of parts verification in such environment. Some models allowed parts to be released early in the manufacturing system (Minifie and Davis, 1990; Brennan and Gupta, 1993; Ho and Carter, 1996; Homem-de-Mello *et al*, 1999) when the lower level parts are available, relaxing MRP dependent demand timing rule within which parts must be released according to its release date if no delay has occurred. Parts verification in terms of parts release timing is found to be the most common error that had been ignored, resulting in unnecessary increased in inventory level.

Some models neither used POR schedule or work-to-list to release order for production nor any BOM to define parts relationships (Matsuura *et al*, 1995; Kanet and Sridharan, 1998). It was also identified that their demand were modelled by arrival of jobs rather than defined by the MPS, relaxing the MRP multi-level dependent demand rule. The effects of these omissions are that the orders that were released and the parts that were processed might not be correct, i.e. linked with a specific customer order. This will in turn give a false performance result on the delivery level of a manufacturing system. Research on how to release orders in such environment has also been carried out, but with little notion on parts verification in the simulation model (Donselaar and Gubbels, 2002).

In this paper, we refer to this environment as a multi-level dependent demand manufacturing system, owing to the nature of parts relationship in an MRP system. To verify that the correct parts are used for the correct order, two levels of accuracy are examined namely: the correct supply of quantity required and at the correct timing for release. A recognition and classification structure is designed and implemented to execute parts verification in a multi-level dependent demand manufacturing system. The following sections discuss the developmental work of the structure, verification and validation simulation experiments results and analysis, and the robustness of the structure.

2. A recognition and classification structure for parts verification

Parts verification can be defined as a method to check the accuracy of the release of a work-to-list in relation to its planned quantity required and its timing. This research has designed and implemented a recognition and classification structure to execute parts verification in a multi-level dependent demand manufacturing system.

2.1. A part classification structure

In a commercial MRP system, the release of a part can be verified through the use of parts' where-used information. In a simulated MRP model, the parts' where-used is not obvious. Customer order number gives the first level of parts verification, which indicates a part is made/purchased for a particular customer order. Nevertheless, relationships between the parts in/between BOM needs to be explicitly defined to ensure that the correct parts are used for the correct order. Due to the feasibility that a parent part can also be a child part for another parent part, this aspect of multi-level pegging should not be overlooked. To this end, a classification structure is designed to uniquely classify parts into parent and child. Four classes are designed, namely *part tag, parent tag, child count and holding number*. The part classification structure is developed using Visual Basic Application (VBA). The part classification structure is implemented in the simulated MRP model, which was developed in the previous research (Koh and Saad, 2002a). The implementation of the part classification structure will provide a correct and verified work-to-list, particularly in terms of the required quantity and the release timing, for parts that are multi-used within/between product(s). Figure 1 shows an example of a work-to-list that is generated from the implementation of the part classification structure from the simulated MRP model.

[Insert Figure 1 about here]

Part tag is a concatenation of customer order number and part number. This gives a unique identity to claim a specific part for a specific order. At the initial stage of creating the part tag, let suffix *i* = a part, which could be a parent part *j* or a child part *k*. This can be written as:

$$p_i \equiv q_i \text{'\&'} r_i \qquad (1)$$

with p_i = *Part tag* of part *i*

q_i = Customer order number of part *i*

'*&*' = Concatenation symbol in Excel/VBA

r_i = Part number of part *i*

To enable a child part to recognise its parent part, the *parent tag* is designed. Every part has a *parent tag* and the *parent tag* is a reproduction of the part's parent's *part tag*. This reproduction logic allows a child part to accurately match with its parent part. Let suffix *j* = a parent part, *k* = a child part and *l* = level in the BOM. This can be written as:

$$s_{k,l+1} \equiv p_{j,l} \qquad (2)$$

with $s_{k,l+1}$= *Parent tag* of a child part *k* at level *l+1*

$p_{j,l}$ = *Part tag* of a parent part *j* at level *l*

Child count defines the number of required child part at the next level down the BOM chain. The main goal of this class is to verify a state within which all required child parts are completed and available for the production of a parent part. The computation for *child count* can be written as:

$$n_i = \sum z_i \tag{3}$$

with n_i = *Child count* of part *i*

z_i = Part at the associated level down the BOM chain of part *i*

To verify multi-level dependent demand between parts in a finite-capacitated manufacturing system, the release of all parent parts have to be constrained so that the dependency with its child parts can be accurately modelled. The main goal of this constraint is to prevent early or incorrect release of order. Based on our part classification structure, we are able to recognise which is parent part in the work-to-list. This is simply denoted by any part in a work-to-list, which has a child count greater than zero. This condition is used to identify all parent parts in the work-to-list and assign a constraint to them. The constraint is initiated by a second sequence number, which will direct these parts to a predefined queue in the multi-level dependent demand manufacturing simulation model and being held in the queue to wait for their child parts to arrive. We called this the *holding number*. To coordinate parent part searching, every parent part in a product is assigned with the product's part number as their *holding number*. This is found to have significantly reduced computational time for the simulation run. Let suffix *f* = a finished product. This classification can be written as:

$$h_j \equiv r_{f,j} \tag{4}$$

with h_j = *Holding number* of a parent part *j*

$r_{f,j}$ = Part number of a finished product *f*, which uses the parent part *j*

This part classification structure is extended to include the multi-use of a common part. In this case, the part number of the parent part is concatenated with the part number of a child part to form a second level unique identity for the *part tag* of the child part. Since the *parent tag* of the child part will be assigned according to the way described above, the common child part will always be allocated with the correct parent part and this will guarantee the correct part is used for the correct order.

2.2. A part recognition algorithm

In a finite-capacitated manufacturing environment, the work-to-list may not be valid when delay is encountered. Moreover, a manufacturing system simulation model by nature is a forward executing environment within which dependent demand is not automatically considered, but on the other hand it will process any part that are available at the resources according to the queuing rules specified. To model a multi-level dependent demand manufacturing system under these conditions, we design a part recognition algorithm and implement the algorithm in a simulated multi-level dependent demand manufacturing system. The main goal of the part recognition algorithm is to identify the correct parts to be used for the correct subassembly and finished product in a finite-capacitated multi-level dependent demand manufacturing system. The multi-level dependent demand manufacturing simulation model is modelled using ARENA simulation software and the parts recognition algorithm is coded in the model using SIMAN simulation language. SIMAN is the language that codes the simulation model in ARENA (Pegden *et al*, 1995; Kelton *et al*, 1998).

In the part recognition algorithm, we program and initialise a condition – *child count* greater than zero ($n > 0$), which will recognise all parent parts and route these parts to their respective queues (h). If $n = 0$, this confirms that this part (usually purchased parts) does not depend on any child parts for production and hence it can be routed for supply. $n < 0$ is implausible. The part is in the form of an entity, which is held and to be released from the queue. Their release conditions are twofold: (1) whenever all required child parts are completed and available, and (2) whenever the release time is valid.

The former condition is denoted by a status in the simulation run – $n = 0$. To recognise whether this status has been achieved, we assign n as a variable in the simulation model, which will reduce by one whenever a required child part is available. n is initialised by the summation of part at the associated level down the BOM chain (z). Over time, the value of n will be decrementing and finally it will reach zero. Whenever this status is achieved, it indicates that there are no part shortages for the production of the parent part and the parent part’s release time can be evaluated.

The latter condition is programmed through a series of If-Then-Else expressions. If Time Now (TNOW) (the actual time in the simulation) = release time of the part in the work-to-list, then this part can be released for production at TNOW. If the required child parts are delayed in production, then they will be delivered late. This will make TNOW of the parent part > release time in the work-to-list, and this part will be released late (using TNOW). However, it is possible that the planned release time is over-pessimistic, i.e. greater than it is necessary. The effect of this situation is that the TNOW of the parent part may well be < release time in the work-to-list, and hence early release will happen if an explicit correction is not programmed. This status has to be avoided because continuously releasing parts earlier then necessary will result in unwanted increase in inventory level and ultimately will increase the total cost of production, particularly when increasing value is added to the parts in the production process. To resolve this problem, we program a delay between the release time in the work-to-list and TNOW to minimise this effect, which will enable the parent part to be released at its planned release time.

Above all, it is also important to verify that the correct parent part is released from the queue (h) to the correct child part. Here, we use the *part tag* (p) and the *parent tag* (s) classifications in the work-to-list. They are assigned as an attribute respectively in the simulation model. We have already established that the s of a child part = the p of its parent part. Whenever a part finishes its operations, we assign the s of the part to a variable called completed tag (t). Completed tag denotes a part has completed its operations and it is ready to identify who is its parent part. This variable is designed to improve the part recognition efficiency particularly when several parts are completed at the same time. The assign expression can be written as:

ASSIGN: t = s; (5)

To enable parent part recognition and verification, a search expression is programmed. The search expression can be written as:

SEARCH, h: (t == p); (6)

This code searches the parent part queue (h) to recognise the correct parent part. The finding of the correct parent part is verified by the true condition of $t{=}{=}p$, which signifies that the p of the parent part is equalled to the t of its child part.

Once the correct parent part is found ($s = p$), the queue index of the parent part is then internally assigned to a search index (J). It aims to remove the entity of the succeeded search. The remove expression can be written as:

REMOVE: J, h, CSOP; (7)

This code removes the *J* indexed parent part entity from its queue (*h*) and goes to a program line to check for the status of the parent part (*CSOP*). This check starts with decreasing the *child count* (n) of the parent part by one and then evaluates the new value of *n*. Only when $n = 0$ that the parent part can be considered to be released (subject to the above release time verification subroutine), otherwise the parent part will be re-routed back to its queue (*h*).

The program and the codes have been verified and the part classification and recognition structure clearly shows that: multi-level dependent demand manufacturing system can be accurately modelled and verified. The problem of part shortages at parent part level is eliminated. The release time of parts accurately reflects to a realistic MRP condition.

3. Implementation of the part recognition and classification structure

The part recognition and classification structure is implemented in a real case study. The case company is a medium-sized commercial transformer manufacturer, based in London, UK. They use WinMan system for production planning and scheduling. The implementation is carried out in two stages as shown below:

3.1. Stage 1 - parameterisation

Parameterisation involves populating the simulated MRP model and the multi-level dependent demand manufacturing simulation model with an exact replica of the case’s planning and manufacturing resources data in their WinMan system and the manufacturing system. The planning data is derived from the MPS, BOM, Item Master File (IMF) and Routing File (RF) from the WinMan system, whilst the manufacturing resources data is derived from the physical constituents of their shop floor, such as types of machines, number of machines, shift pattern and queuing rules. The effect of plant layout on the performance of the part recognition and classification structure is not considered in this study. Table 1 shows the parameters in this study.

[Insert Table 1 about here]

A *2* years MPS is considered to be adequate because their products have a relatively short average cycle time (*10* days). Searching through their order book to find a representative mixed demand pattern that consists of varied order size and order time interval led to a clear ten products (multi-products) chosen for this study. A mixed demand pattern is modelled because it has been identified that multi-level dependent demand manufacturing environment with such demand pattern suffers greater level of uncertainty (Koh *et al*, 2000).

A total of *3* runners, *4* repeaters and *3* strangers, classified by their order interval (Parnaby, 1988) expresses the mixed demand pattern. Runners are products that have a regular and stable demand pattern, which have a relatively shorter order interval. Repeaters are products that have a quite regular and stable demand pattern, but with a relatively longer order interval. Strangers are products that have no regular and stable demand pattern, which usually have the longest order interval. The average order size for these products ranges from a minimum of *30* to a maximum of *250* units. They represent a mixture of *60%* purchased parts and *40%* manufactured parts (including the finished product). They have a minimum of *3* levels to a maximum of *5* levels BOM (multi-levels). A total of *434* associated different parts with their lead-times are modelled. Their routings are extracted and a routing number is being assigned. MRP run for these selected MPS has resulted in some *50000* orders in the work-to-list. Finally, the values of *part tag* (p), *parent tag* (s), *child count* (n) and *holding number* (h) for the parts in the work-to-list are assigned.

A total of *16* combinations of machines and labour resources are modelled in the multi-level dependent demand manufacturing simulation model. In SIMAN, these resources can be represented using the *RESOURCES* element and they are coded in the experiment file. The *RESOURCES* element can be written as:

RESOURCES: Name, Capacity; (8)

The capacity of the machine/labour indicates the number of machines/labour and the labour works on an *8*-hours 1 shift for 5 days week. It can be noted that some of the resources do have a relatively high capacity level due to its utilisation by all products range.

Every resource is allocated with a predefined queue, which is associated to its incoming parts and outgoing parts. Earliest Due Date (EDD) queuing rule is set at the resources' *QUEUES* element. This rule is defined in the Low Value First (LVF) command (by due date) in the *QUEUES* element. The *QUEUES* element can be written as:

QUEUES: ResourceQ, LVF(Due Date); (9)

It picks the part, which has the minimum value of due date from the resource's incoming queue for production. There may be several parts in the resources' queues, each has a very close due date. Thus, a *RANKINGS* element is used to prioritise the parts by due date in chronological order. The *RANKINGS* element can be written as:

RANKINGS: Resource_1Q-Resource_nQ, LVF(DueDate); (10)

It is established earlier in this paper that uncertainty in production is inevitable, thus it is important to show that the part recognition and classification structure has been successfully implemented with such consideration. The related work can be found in Koh and Saad (2002b), Koh and Saad (2003a), Koh and Saad (2003b) and Saad *et al* (2003). For the purpose of this study, we made a number of assumptions in the simulation model. Explicit machine failure and labour absentee are not allowed. The only time when the required machine/labour is not available is when they are busy processing other parts. These assumptions have to be made to simplify the implementation process of the part recognition and classification structure. Uncertainty can be easily modelled in the simulation model once the structure is validated.

3.2. Stage 2 - execution

Execution refers to running the case company's multi-level dependent demand manufacturing simulation model using the work-to-list that is developed through the part recognition and classification structure. The first step in the execution process is to segregate the parent part from all other parts. This segregation enables parent part to be routed to its holding queue (*h*). This is modelled using *ROUTE* block in the model file and the routing process is defined in the *SEQUENCES* element in the experiment file. Prior to the routing, the *sequence number* (*NS*) of the entity has to be assigned to *h*. The *SEQUENCES* element can be written as:

SEQUENCES: NS, h; (11)

When the *ROUTE* block is run, the entity will be routed to *h*.

The second step of the execution is to route the child parts at the lowest level of the BOM for supply. A similar *SEQUENCES* element is used as and it is shown below. However, the *NS* for these parts will be assigned to their routing number, which indicates who is the supplier and the purchase lead-time.

SEQUENCES: NS, Supplier name, Lead-time & Exitsystem; (12)

In this case, the routing number of the part takes precedence because its release does not depend on any parts at its lower level. Once the part has arrived, the entity exits the system. The arrival is modelled using the *DELAY* block in the model file and it can be written as:

DELAY: Purchase Lead Time; (13)

The third step of the execution is to run the *SEARCH* and the *REMOVE* programs. Let illustrate this with an example. Assume part number *19009* (a child part) to assemble part number *19* (a parent part) for customer order *19546* has arrived, and the holding number of the parent part is *19*. The below execution will follow:

ASSIGN: t = 1954619;

SEARCH, 19: (t == p);

REMOVE: J, 19, CSOP;

If *p = 1954619* is found, the *J*-indexed parent entity from *h = 19* will be removed and the status of part number *19* is checked. This check executes the following commands:

```
CSOP    ASSIGN:      n = n - 1;
                            IF:     (n==0);
                                    ASSIGN: NS = Routing number;
                                    ROUTE;
                            ELSE;
                                    ROUTE;
                            ENDIF;                                    (14)
```

If a total of *7* child parts are required to make part *19,* the completion of part number *19009* will result in *n = 6.* Part number *19* will be re-routed back to queue *19*. Only until *n = 0*, the manufacturing routing number of part number *19* will be assigned to its *NS* for manufacture.

The fourth step of the execution is to route the parent part for manufacture when the above condition (all required child parts are available) is met and when the release time is valid. Manufacture for parent part is processed through a number of resources. The routing number directs the parent part to the assigned resources with reference to its operations sequence. The operations sequence is stored in the *SEQUENCES* element, which tells us - which are the assigned resources, the part's set-up time and operation times. Once the part is manufactured, the entity exists the system. The *SEQUENCES* element can be written as:

SEQUENCES: NS, Resource_1 station name, Set-up time & Resource_n station name, Set-up time, Operation time & Exitsystem; (15)

The parts are processed in batches. The larger is the net requirement (batch size), the longer it will take to process the batch. This is modelled using the *DELAY* block and it is written in the model file as follow:

DELAY: Set-up Time + (Net Requirement x Operation Time); (16)

Whenever a part is available or completed its operations, the third step of the execution is recurred. This recursion ensures that all parts in a BOM are processed particularly for the parent part that is also a child part.

The fifth step of the execution is to timely release the parent part for manufacture. In ARENA, whenever an entity is released, TNOW is used as the default. This feature is exploited to our advantage in terms of on-time release and late release because TNOW will be valid. However, to prevent early release, a *DELAY* block is used to offset the earliness. This will result in the planned release date being used instead of TNOW. The condition can be written as:

```
IF: (Release Date > TNOW);
        DELAY:        Release Date – TNOW;
ENDIF;                                                  (17)
```

4. Results, analysis and discussions

Simulation experiments are carried out to verify the internal validity and logic effectiveness of the part recognition and classification structure. To examine the external validity of this structure, the simulation results are validated against the outcome generated from a commercial MRP system.

In the verification process, the accuracy of the values of the attributes for each entity in the work-to-list is checked, especially the *part tag* (*p*), *parent tag* (*s*), *child count* (*n*) and *holding number* (*h*), prior to reading by the multi-level dependent demand manufacturing simulation model. An *ERROR* variable is coded in the simulation program and will be returned if there is any loose link between parent part and child part (incorrect *p* and *s*). No *ERROR* has been found. To validate this, a replica of the data in the simulated MRP model is set-up in a commercial MRP system (Alliance Manufacturing Software). This enables comparison between the work-to-lists. Identical release dates and net requirements are achieved. The where-used feature in Alliance Manufacturing is used to find the match between parent part and child part. We found an exact match for all parent part and child part that are linked by the use of *p* and *s*.

Simulation stepping is also performed to monitor the sequence of events during the simulation run. This verifies whether *child count* (*n*) reaches zero before the parent part is released. At all release time, the *n* value of the parent part is saved into a file. The file shows that all parent part has $n = 0$. This confirms that no part shortages occurred in the simulation run. The simulation stepping also watches whether the correct release time is used at each release.

We use a numerical example to show the results. Figure 2 shows a simplified BOM for product number 11 with consideration of some common parts. Let due date for this product = minute *40500*. Table 2 shows the verified output.

[Insert Figure 2 about here]

[Insert Table 2 about here]

The first operations for all child part of part number *11001* were started at minute *0*. However, the first operations for all child part of part number *11003* were started *10* minutes later. Part number *33001* for use by parent part number *11002* was also started at the same time. After another *10* minutes, part number *33004* for use by this parent part started its first operation. Since the first operations for part number *33002* and *33003* for use by part number *11001* was completed at minute *2500*, their second operations started at this time. Adhering to this logic, the subsequent operations for a part are determined. Once a child part has completed all its operations, its completion time is recorded in the child part completion section. Its due date is then compared with its completion time to identify whether it is being delivered on time. This section shows that all of the parts were delivered early, except for part number *11001*, which was delivered late by approximately *5* days.

Stepping through the simulation run continues until a plausible release is found for a parent part. The plausibility is determined at the time that its last required child part is completed. This implies that the child part's latest completion time is plausible to be its parent part's release date. This information is recorded as *TNOW* in the plausible parent part release section. Referring to the first parent part release section, release date for part number *11001 = 28800*, which has not been reached because *TNOW = 6000*. Therefore, part number *11001* was not released at this time.

In this case, the simulation will continue to process other parts until all required parts are made. If the release date has not been reached at this point, a delay of the difference between the release date and *TNOW* will be executed in order to move the simulation time forward so that early release can be avoided. This delay applies to parent part, which is yet to be released, based on a chorological order of the release date. Hence, part number *11001* was released last, just before the finished product started its operations. Its longest completion time has resulted in a late release for its parent part, part number *11* by approximately *5* days. However, knock-on effect to its parent part's delivery has not been found, instead its parent part was delivered early. This may be due to the safety lead-time in the IMF module or resource overload.

The numerical example has shown parts' early and late deliveries. This type of verification output is generated for all entities in the work-to-list. The results have proved the internal validity of the part recognition and classification structure for parts verification in a multi-level dependent demand manufacturing system. To validate this, the parts' earliness and tardiness are built into their lead-times before MRP run in Alliance Manufacturing. This gives a similar result to the verification output. Nevertheless, there are some anomalies, which is when lead-time is believed to be accurate but yet parts are still being delivered late.

It is hypothesised that this late delivery is due to resources overload. To this end, a mapping analysis is carried out to identify the parts involved and timing of the late delivery. For each part identified, its routing is uncovered, giving a total of 5 resources that plausibly cause the late delivery. These resources are those with the highest overall utilisations. A spreadsheet is then devised to analyse the late delivery against resources utilisations given by Alliance Manufacturing. A five-day moving average for the resources utilisations is calculated to reflect a typical industry planning time bucket. Table 3 shows the results of the mapping analysis.

[Insert Table 3 about here]

The results broadly support the hypothesis that the late delivery occurs in periods when there are a number of consecutive days of resources overload. The late delivery resembles with the simulation output. In the absence of an MRP system that includes finite scheduling, these results provide the highest level of validation possible.

Simulation stepping is also used to validate the routings. An exact match has been found for the routings through comparison between the operations sequence in the simulation run with the routing in Alliance Manufacturing.

The verification and validation results show that the part recognition and classification structure has been proved to be successful for parts verification in a multi-level dependent demand manufacturing system. It has been successfully implemented in our previous studies from the uncertainty diagnosis perspective (Koh and Saad, 2002b; Koh and Saad, 2003a; Koh and Saad, 2003b; Saad *et al*, 2003).

5. Conclusions and implications

The main contribution from this work is the development and the implementation of a part recognition and classification structure for parts verification in a multi-level dependent demand manufacturing system. This structure has successfully filled the gap that little effort was emphasised on parts verification to ensure that the correct parts are being produced and allocated to its specified demand. We showed that the implementation of this structure ensures customer order will not be compensated with other orders.

The development of the part classification structure enables the modelling of a multi-level dependent demand manufacturing between parts to be carried out effectively. This was achieved through the design of the four new classes, namely *part tag*, *parent tag*, *child count* and *holding number*, assigned to each entity in the work-to-list before releasing this list to the multi-level dependent demand manufacturing simulation model. The part recognition algorithm enables the parent and child relationship between parts to be recognised in a finite-capacitated manufacturing system. This was achieved via programming the classification structure into the simulation environment and executing the codes under pre-defined constraints, which were coded in a series of conditional expressions. Part release was prohibited at three levels: incorrect parent and child, part shortages, and invalid early release.

The part recognition and classification structure was successfully implemented using a real case study. This was achieved through two stages: parameterisation and execution. Parameterisation involves populating the simulated MRP model and the multi-level dependent demand manufacturing simulation model with an exact replica of the case's planning and manufacturing resources data in their WinMan system and the manufacturing system. In the execution stage, five steps were involved.

Step 1: Parent parts were segregated from other parts

Step 2: Child parts at the lowest level in the BOM were routed for supply

Step 3: Child part that has completed its operations was commanded to search its parent part

Step 4: Parent part was routed for manufacture

Step 5: Parent part was released for manufacture according to the specified conditions

Verification and validation of the part recognition and classification structure were successfully performed. The results were shown using a numerical example of a simplified BOM. Simulation stepping was mainly used for this verification. Results from Alliance Manufacturing were used for validation. In both cases, we found that the output supports each other. We showed that when lead-time is considered to be accurate, late delivery could cause by resources overload.

The results led to the conclusion that implementation of the part recognition and classification structure has effectively verified the correct parts and sub-assemblies used for the correct product and order. No parts and sub-assemblies shortages were found, and the quantity required was produced. The scheduled release for some orders was delayed due to overload of the required resources. When the loading is normal, all scheduled release timing is adhered to.

This structure has a robust design; hence it can be easily adapted to new systems parameter to study a different or more complex case. This structure could act as a template for researchers and practitioners in this field to examine various operational and managerial issues in a multi-level dependent demand manufacturing system that is controlled by MRP rules.

References

Brennan, L. and Gupta, S.M., 1993, A structured analysis of material requirements planning systems under combined demand and supply uncertainty. *International Journal of Production Research*, vol. 31, no. 7, pp. 1689-1707.

Donselaar, K..H. van and Gubbels, B.J., 2002, How to release orders in order to minimise system inventory and system nervousness?. *International Journal of Production Economics*, vol. 78, no. 3, pp. 335-343.

Ho, C.J. and Carter, P.L., 1996, An investigation of alternative dampening procedures to cope with MRP system nervousness. *International Journal of Production Research*, vol. 34, no. 1, pp. 137-156.

Homem-de-Mello, T., Shapiro, A. and Spearman, M.L., 1999, Finding optimal material release times using simulation-based optimisation. *Management Science*, vol. 1, no. 45, pp. 86-102.

Kanet, J.J. and Sridharan, S.V., 1998, The value of using scheduling information in planning material requirements. *Decision Sciences Journal*, vol. 29, no. 2, pp. 479-498.

Kelton, W.D., Sadowski, R.P. and Sadowski, D.A., 1998, Simulation with ARENA, McGraw-Hill.

Koh, S.C.L. and Saad, S.M., 2002a, Design and implementation of ERP-controlled manufacturing in simulation. Proceedings of the 2nd International Conference on Responsive Manufacturing, Gaziantep, Turkey, pp. 345-350, 26-28 June.

Koh, S.C.L. and Saad, S.M., 2002b, Development of a business model for diagnosing uncertainty in ERP environments. *International Journal of Production Research*, vol. 40, no. 13, pp. 3015-3039.

Koh, S.C.L. and Saad, S.M., 2003a, A holistic approach to diagnose uncertainty in ERP-controlled manufacturing shop floor. *International Journal of Production Planning and Control*, vol. 14, no. 3, pp. 273-289.

Koh, S.C.L. and Saad, S.M., 2003b, MRP-controlled manufacturing environment disturbed by uncertainty. *International Journal of Robotics and Computer Integrated Manufacturing*, vol. 19, no. 1-2, pp. 157-171.

Koh, S.C.L., Jones, M.H., Saad, S.M., Arunachalam, S. and Gunasekaran, A., 2000, Measuring uncertainties in MRP environments. *Logistics Information Management: An International Journal*, vol. 13, no. 3, pp. 177-183.

Koh, S.C.L., Saad, S.M. and Jones, M.H., 2002, Uncertainty under MRP-planned manufacture. *International Journal of Production Research*, vol. 40, no. 10, pp. 2399-2421.

Matsuura, H., Tsubone, H. and Kataoka, K., 1995, Comparison between simple infinite loading and loading considering a workload status under uncertainty in job operation times. *International Journal of Production Economics*, vol. 40, no. 1, pp. 45-55.

Minifie, J.R. and Davis, R.A., 1990, Interactions effects on MRP nervousness. *International Journal of Production Research*, vol. 28, no. 1, pp. 173-183.

Parnaby, J, 1988, A system approach to the implementation of JIT methodologies in Lucas Industries. *International Journal of Production Research*, vol. 26, no. 3, pp. 483-492.

Pegden, C.D., Shannon, R.E. and Sadowski, R.P., 1995, Introduction to Simulation Using SIMAN, 2nd edition, McGraw-Hill.

Saad, S.M., Koh, S.C.L. and Gunasekaran, A., 2003, Tackling uncertainty in ERP-controlled manufacturing environment: A knowledge management approach. Proceedings of the 1st International Conference on Performance Measures, Benchmarking and Best Practices in New Economy, Guimaraes, Portugal, pp. 622-629, 10-13 June.

List of figures and tables