Proceedings of the 6th World Congress on Intelligent Control
and Automation, June 21 - 23, 2006, Dalian, China

# Modeling of Common Organizational Structure for Reconfigurable Assembly System [*]

Fanli Meng
*Shenyang Institute of Automation (SIA), CAS*
*Graduate School of the Chinese Academy of Sciences*
*Shenyang Liaoning Province, P.R.China*
meng.fanli@sia.cn

Dalong Tan and Yuechao Wang
*Shenyang Institute of Automation (SIA), CAS*
*Chinese Academy of Sciences*
*Shenyang Liaoning Province, P.R.China*
{dltan & ycwang}@sia.cn

***Abstract* - In order to response to the changing market and overcome the disadvantage of the traditional assembly line, based on decomposition of a product family into several so-called Functional Objects, a common organizational structure for Reconfigurable Assembly System is proposed, which consists of truck line and assembly modules. This self-similar organizational structure has many advanced characters such as capacity flexibility, robustness, scalability and so on. Using CPM/PERT, a scalability algorithm is proposed and a scheduling approach is also available to realize robustness of the assembly system.**

***Index Terms* - *RAS; open organizational structure; CPM-PERT; scalability; robustness***

## I. INTRODUCTION

New technological developments and market demands have major impacts on manufacturing. As a result, focus of manufacturing processes has been continuously shifting. Until 1970's, the aim was increasing production with few product variations. 80's saw the dominance of cost reduction and improved quality with some product variety [1]. More recently (starting in the 1990's), the focus has shifted to responsiveness of the manufacturing system, and the manufacturing environments have been getting no more static but dynamic. That demands a responsive manufacturing system that can be rapidly designed and has the ability to adapt new product models produced in unpredictable quantities. For those reasons, a novel manufacturing system, Reconfigurable Manufacturing System (RMS) [2] was proposed by Koren in 1999. So far, RMS has been widely studied. Popa [3] described the methodology used to design and build a reconfigurable micro-assembly system for Photonics applications, and they used the over control scheme and employed the modular architecture to the system. Kong [4] presented an integrated approach for rapid reconfigurable fixture deployment based on fixture workspace synthesis and fixture visibility analysis to produce a family of parts. It is a pity that there is not a common structure for Reconfigurable Assembly System (RAS) up to now. So the theme of this study is to develop a common structure for RAS, which is designed for the assembly of a product family and also can response to continuous changes in the assembly system.

The remainder of this paper is organized as follows: in the section II, how to decompose a product family into so-called functional objects is narrated, and the definition of functional objects is also given. The common organizational structure for RAS is proposed in the section III. In the section IV, CPM/PERT is introduced briefly. In the section V and VI, the optimization algorithm about scalability and robustness is dealt with in detail. Conclusions are drawn in the section VII.

## II. PRODUCT FAMILY REPRESENTATION

Reconfigurable Assembly System (RAS) is evidently designed for a large variety of products to meet market demands. If every different product is studied as an individual, it is very complicated to design the whole assembly system. So the concept of product family (PF) is proposed to reduce the complexity. PF is either product related, putting the accent on a commonality of the constituents, or process related, focusing on the processes and technologies used to produce the products. Product variant (PV) is another concept which is directly derived from the definition of a PF. We adopt the approach of McKay [5], a variant of a PF is an individual product that belongs to this PF. We decompose the PF into so-called functional objects (FOs). So the family is then considered as an assembly of FOs which are designed and analyzed separately.

It is a clear fact that every product variant in a product family consists of several corresponding common parts. Those parts are always similar except that their sizes, shapes maybe are different, or a part of one PV has more auxiliary components than that of another one. Although there is a little of differences between common parts of product variants, it is reasonable to study the common parts as objects which describe things of the same class. Hence we introduce the concept functional objects (FOs) which are equivalent to objects in software. We can get a generic model of a PF by using FOs mentioned above, and this model will produce an imaginary product. An effective representation of a PF can only be obtained through its separate variants. When dealing with a PF, we have to think at a higher level of abstraction, and decompose the family into the relevant functional objects. Without functional objects, the decisions on the product and the process would be made on the whole PF, and incidentally through all its PVs. In the proposed idea, a PF can be analyzed

[*] This work is supported by National Basic Research Program (973 plan) #2002CB312200.

FO by FO, which makes easier for the designer to check the character of every FO.

Now we give the definition of functional object (FO). Product family is noted as F, and functional object as which is a set of generic components (GCs) of F and possibly one imaginary part, so that [6]:

a. The FO is a generic functional constituent of F;

b. Two FOs cannot comprehend the same GC,

$$\forall i, j \in \{1, ..., nFO\} \quad \mathrm{FO}_i \cap \mathrm{FO}_j = \Phi \quad \text{if} \quad i \neq j$$

c. The union of all FOs must be equal to the union of the set of GCs in F,

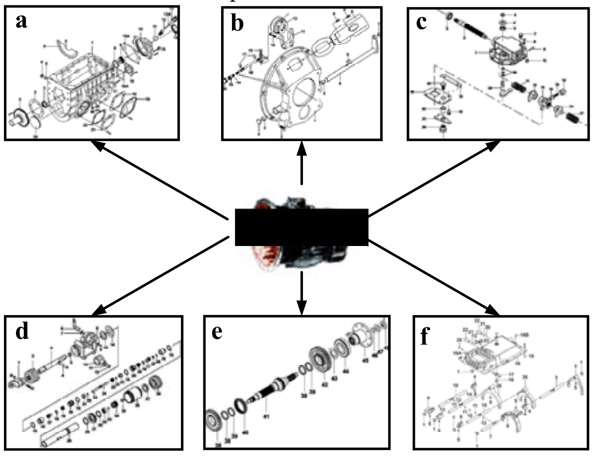
$$\bigcup_{i=1}^{nFO} \mathrm{FO}_i = \bigcup GC_F$$

where $nFO$ is the number of FO in the PF;

d. The choice of the FOs allows the assembly of each PV in F as an assembly of FO variants.

The decomposition of a PF into FOs induces a partition of the set of its components, and a PV is seen as an assembly of FO variants. Every FO is independent, which means that it could be assembled on an independent workstation. This idea is close to the "chunks" introduced by Ulrich[7]. After decomposition, we introduce another important concept, base-part, which is one of the FOs. Base-part is such a FO that most of the other FOs will be attached to it. The purpose of base-part in PF will be explained in the section III.

We derive the common structure for RAS by designing of the assembly system of a gear-box product family. The gear-box product family is decomposed into six functional objects (FOs) including the reduction case (a), the clutch shell (b), the roof cover (c), the strengthener (d), the gear shaft (e) and the upper cover (f) as shown in Fig.1, and the reduction case is taken as the base-part.



[redacted]

Fig.1 Functional objects of the gear-box product family

## III. Common Organization Structure

RAS has been studied widely in terms of many aspects, but up to now, there is no verdict about what on earth the system looks like, i.e. what is the common structure for RAS. In this section we will give our viewpoint of the common structure of RAS in terms of organization as shown in Fig.2.

The whole assembly line is decomposed into trunk line and assembly modules. This structure is capable of facilitating reconfiguration of assembly system. When the assembly system needs to be altered to response to the market, it doesn't need to change entirely, and the same goal can be attained by some local modification of several assembly modules. This structure overcomes the shortcomings of traditional assembly systems such as linear organization, obligatory process flow and so on, and at the same time increases the flexibility, scalability and robustness.

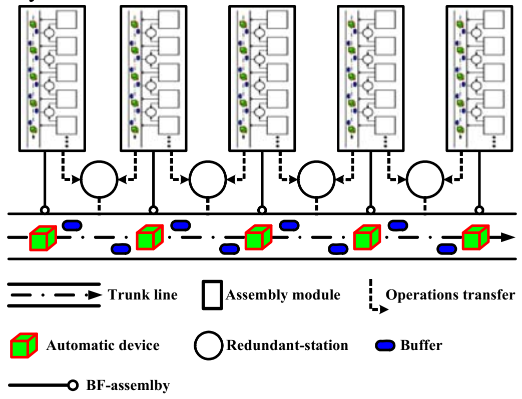


Fig.2 Common organizational structure of RAS

### A. Trunk Line

Trunk line is also called main line of RAS. Starting with the base-part of a product family, the remaining functional objects (FOs) being attached at the various assembly sites the base-part visits. Trunk line has an alterable configuration, in which some automated devices are taken as "process signs" and the number of workstations can change according to the cycle time. "Process signs" means those automated devices keep immovability after reconfiguration of the system. In a product family, the corresponding assembly operations in trunk line are always similar, and the assembly plan is also similar, so some necessary automated devices are necessary for all the variants in the family and become "process signs".

Trunk line is the active section in RAS, deciding the cycle time of the whole system. The assembly modules adjust their configurations to meet trunk line in order to ensure their operation times will not exceed the cycle time.

At the beginning, we define two assembly types:

BF-assembly: assembly between base-part and functional objects;

FF-assembly: assembly between functional objects.

It is clear that BF-assembly is completed in trunk line of RAS. In our study, according to the character of the gear-box product family, manual operations and automated operations exist in trunk line.

Trunk line has the following advantages with some additional instruments.

*1) Capability Flexibility*: the ability of trunk line to adapt to all the variants in a product family. The capability flexibility is realized by the corresponding automatic devices, which means these devices have enough flexibility for the whole product family in terms of preciseness and performances. In RAS for the gear-box product family, the automatic devices include pressure machines, glue spreaders, gaskets selecting machines, turnover rigs, bolts fastening machines and so on. Through parameters adjustment and modification of NC codes, the automatic machines can assembly all the variants in the gear-box family. The assembly devices of trunk line should be placed in the proper position along trunk line, the locations are determined according to the base-part, and some of these devices are taken as "process signs". AGVs (Automatic Guided Vehicle Systems) are utilized for the material handing system in trunk line. The layout of trunk line is similar to the traditional stream line.

*2) Process Robustness*: the ability of anti-disturbance. In the traditional assembly line, if a station breaks down, soon stations up the line have to stop because base-parts cannot proceed, and stations down the line also have to stop operation by reason of the supply shortage. We introduce two means to increase the robustness of trunk line.

Firstly, based the on classical layout, trunk line is extended by flexible buffers. These buffers are located along trunk line and can be used as additional buffering capacities if a disturbance happens. If the broken device can be repaired in time, the base-parts in the buffers will be transported back to trunk line and the assembly system will resume normally. Because these buffers exist, the stations in front of disturbance is able to continue processing as long as there is room available in the buffers, and the stations after the disturbance are able to continue processing as long as there are base-parts available in the buffers.

Secondly, we change the flow process to increase robustness of trunk line. It is clear that adding buffers into trunk line is very useful for resisting short-term disturbances. If the broken device can't be repaired in time, buffers will lose function. In that case, we find the "substitute device" in the other assembly modules which can complete the same assembly operation. Then we change workflow, and transport the base-part to the corresponding assembly module in which the "substitute device" exists. After assembled by the "substitute device", the base-part is transported back to the original station, so it can proceed to the next stations. It must be emphasized the fact that special tracks and necessary interfaces are equipped between the same devices which are in the different place. Those special tracks and the track of trunk line never cross each other. Henceforth, this "substitute" approach will not confuse the assembly system. After returning to the original station, the base-part is assembled just as in trunk line. It can be seen that inflow buffers and outflow buffers of the "substitute devices" must be flexible. When designing those buffers, we must consider not only the part type of upstream-stations, but also of other modules.

*3) Scalability Flexibility*: the ability of increasing the throughput to respond to market demands in terms of quantities. The classical layout is extended by so-called redundant-station which derives from the idea of Bussmann [8]. But our proposed redundant-stations are very different. A redundant-station is a "cross-processing" station, and able to perform partial manual operations of two actual adjacent stations in trunk line. The redundant-stations are often regarded as an imaginary station and useless without scalable demands. When the market demands much more products, the redundant-stations will manifest their prominent advantages. The scalable reconfiguration of trunk line is a process of transferring some of operations from original stations to the redundant-stations.

At the beginning, we delete some operations in two original adjacent stations, and add these operations to the corresponding redundant-station. If every station on the trunk line has an accessorial redundant-station, after transfer of operations, the cycle time of trunk line is reduced greatly, because the number of stations in trunk line increases. Adding redundant-stations seems to increase cost of assembly system, but in fact these operations are usually manual and we only transfer operations and increase more workers, so increasing cost is not excessive actually. In some cases, if demand of the product family tends to rapidly increase, the company has to buy new assembly devices to compensate a bottleneck in trunk line and to shorten the cycle time of the assembly system. The extreme case is that every redundant-station has the same overall capacity as original stations in trunk line, so the entire second line of assembly is formed and productivity is doubled as a result.

## B. *Assembly Modules*

Assembly modules in RAS are responsible for assembling the independent functional objects in a product family, i.e. to accomplish F-assembly. Assembly modules are passive, which modify their configurations according to the cycle time of trunk line and determine the interface with trunk line.

The proposed functional object (FO) is an independent part in a product family, so it absolutely can be an imaginary sub-product which has the similar characters as high-level product. Therefore we can analyse a functional object (FO) in the similar manner as we analyse the whole product family. As a sub-product, a FO can consequently be decomposed into sub-FOs, for example, a gear shaft is decomposed into several sub-FOs as shown in Fig.3. In the same way, an assembly module consists of sub-trunk-line and sub-modules as well. Sub-trunk-line is responsible for assembling sub-FOs onto the sub-base-part, and sub-modules have the responsibility for every independent sub-FO. That is to say that every assembly module has the same structure as high-level assembly system, consisting of sub-trunk-line and sub-modules.

Assembly modules have other characters besides capacity flexibility, robustness and scalability flexibility.

*1) Autonomy*: is to say that every module is distributed and autonomous, and has the independent ability to assembly an integrated FO. Every module has its own scheduling plan, arrange the order of different variants into production, and determine how many workstations to satisfy the cycle of trunk line. Assembly modules can also modify their configurations by themselves without other interferes.

*2) Responsiveness*: the ability of rapidly reacting to the dynamic market in terms of the type of variants in a product family. In other words, assembly modules can be adjusted to respond for the variants' changing. Because a new variant is always different in terms of some FOs, the reconfiguration process is to adjust the corresponding assembly modules, so adjustment is local and the other portions of RAS are not needed to change. So the reconfiguration process is quite easier and faster, and the responsiveness of RAS is increased greatly.

*3) Self-similar*: every assembly module has the same layout as the entire RAS, including sub-trunk-line and sub-modules. The self-similar structure facilitates the construction process of modules when we design RAS.

In a word, the common organizational structure enables RAS to realize reconfiguration more easily, and at the same time robustness, scalable flexibility and capacity flexibility are available.

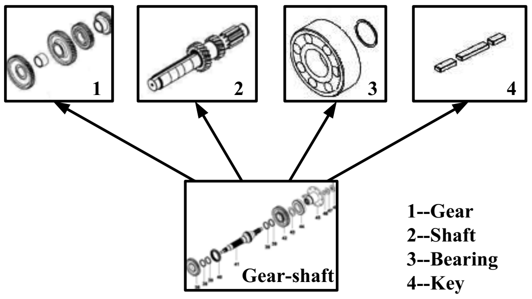


Fig.3 Decomposition of a FO into sub-FOs

## IV. CPM/PERT

CPM/PERT (Critical Path Method/Program Evaluation and Review Technique) is well-known in Engineering Management, Production and Operations Management, originating from America in 1950's. CPM/PERT computations start with a directed network diagram that shows the precedence relationships between the various required activities. Then, a procedure is used to find the key project parameters such as the critical path, the minimum completion time, and the various slack times [9].

This technique is very useful for RAS. This section we will introduce CPM/PERT briefly. In order to calculate all kinds of time parameters, the following symbols are introduced:

ET: the earliest completion time of node;
ES: the earliest start time of activity;
EF: the earliest finish time of activity;
LT: the latest completion time of node;
LS: the latest start time of activity;
LF: the latest finish time of activity;
S: the start time of project (S=0);
T: the calculated completion time of project;
D: time limit for project;
ij: activity;
i: start node of activity (ij);
j: finish node of activity (ij);
$t_{ij}$: duration of activity (ij).

Using the forward pass method, the values of all kinds of earliest times can be obtained with the following formulas:

$$ET_S = S = 0 \tag{1}$$

$$ET_j = \max(ET_i + t_{ij}) \tag{2}$$

$$ES_{ij} = ET_i \tag{3}$$

$$EF_{ij} = ET_i + t_{ij} \tag{4}$$

Note that the earliest completion time of a node is the maximum completion time of all the predecessors. The earliest completion time of the last node is the minimum time for completing the project (T), also is the length of the critical path. With the value of T, the latest times can be calculated by a backward pass method. The correlative formulas are as follows:

$$LT_T = D = T \tag{5}$$

$$LT_i = \min(LT_j - t_{ij}) \tag{6}$$

$$LF_{ij} = LT_j \tag{7}$$

$$LS_{ij} = LT_j - t_{ij} \tag{8}$$

After determining all kinds of the earliest times and latest times, the critical path can be identified by connecting all activities that have zero slack times.

## V. Scalability Algorithm

### *A. Scalability Algorithm*

Scalability flexibility is a prominent character of RAS as mentioned above. In order to increase throughput to meet market demands, we must reconfigure the assembly system. The reconfiguration process is transferring operations from original stations to the redundant-stations. In this section, using CPM/PERT, a scalable reconfiguration algorithm about is proposed. Trunk line is illustrated to explain the algorithm. Firstly, the network diagram of trunk line must be established. Then use the formulas in the section IV to determine the critical path and critical activities.

The following symbols are used in the scalability algorithm:

$A_i(a_{i1}, a_{i2}, ..., a_{ini})$: The critical activity in trunk line, where $i = 1,2,..., m$; m represents the number of stations except redundant-stations; $ni$ is the number of operations in station $i$;

$T(a_{ini})$: The duration of operation $a_{ini}$;

$T(A_i)$: The duration of the critical activity $A_i$;

$CT$: The determined cycle time in order to meet market demands;

$\max(ni) = N$.

The concrete scalability algorithm is as follows:

Setp1. Calculate $\sum_{i=1}^{m} \frac{1}{m} T(A_i)$, and if $\sum_{i=1}^{m} \frac{1}{m} T(A_i) > CT$, stop (the system will never satisfy the determined cycle time) ; else go to step 2;

Step2. Examine the duration of all the activities in trunk line, and find the maximum value, the corresponding activity is $A(a_1, a_2, ..., a_k) \in A_i$ . By definition, $T(A) = \sum_{j=1}^{k} T(a_j)$ is given. If $T(A) \le CT$ , stop; else go to step3;

Step3. Estimate the number of operations in A; if $k = 1$, stop(the system will never satisfy the determined cycle time); else $k > 1$, go to step4;

Step4. Judge the position of A in the critical path, if A is the first activity, go to step6; else if A is the last one, go to step7; else go to step5;

Step5. Assume A' is the predecessor redundant-station of A, and A'' is the successor redundant-station of A. Compare the values of $[T(A') + T(a_1)]$ and $[T(A'') + T(a_k)]$ ; if $[T(A') + T(a_1)] > [T(A'') + T(a_k)]$, then go to step 6; else go step7;

Step6. Extract the last operation from A, and put it into the successor redundant-station A'', then $T(A) = T(A) - T(a_k)$, $T(A'') = T(A'') + T(a_k)$; if $T(A) \le CT$, go to step8; else return step5;

Step7. Take out the first operation from A, and put it into the predecessor redundant-station A', then $T(A) = T(A) - T(a_1)$, $T(A') = T(A') + T(a_1)$; if $T(A) \le CT$, go to step8; else return step5;

Step8. If $\forall T(A_i) < CT, (i = 1,2,..., m)$ , then stop; else return step2.

Infinite loop exists in the above algorithm, so a termination condition is given to stop the program, iterative frequency satisfying $num < m \times (N - 1)$. The symbols m and $N$ are explained at the beginning of this section. The resultant configuration of system maybe is suboptimal, but it is capable of meeting the shorter cycle time and increasing the throughput as well.

## B. Experimental Case

In our study, the reconfigurable assembly system of gear-box product family is an experimental instance. The original network diagram of trunk line is established as shown in Fig.4. The arrow lines represent the corresponding critical activity (i.e. original stations), and the shadow nodes denote redundant-station. The numbers on the arrow line is the durations of the corresponding stations.

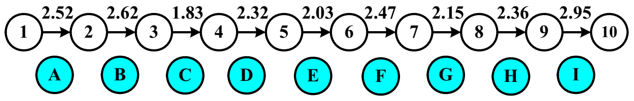


Fig.4 Original network diagram of trunk line

It is can be seen from Fig.4 that the original cycle time of trunk line is CT'=2.95 (min). Assumed market demands more products, the required cycle time is determined as CT=2.5 (min). Using the scalability algorithm, the network diagram of the new configuration is obtained as shown in Fig.5.

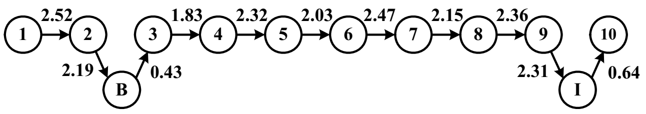


Fig.5 Network diagram of new configuration

After reconfiguration, the first two operations in the activity (9, 10) is transferred into the redundant-station I, and the last operation of (2, 3) is transferred into the redundant-station B. As a result, two redundant-stations are transformed into two actual assembly stations. However, the activity (1, 2) includes only one operation, so it can't be decomposed and transferred to the redundant-station. Consequently, the actual cycle time after reconfiguration is 2.52 (min). In order to satisfy the required cycle time CT=2.5 (min), we must compress the activity (1, 2) compulsively and shorten its duration, and the relevant means is adding new automatic devices and workers into the corresponding stations.

Assembly modules are the passive parts in RAS as mentioned above. According to the network diagram of trunk line, the latest completion times of nodes, which represent the interface between trunk line and modules, are calculated at the beginning, and then the corresponding modules take these values as the time limit of their own F-assembly. In the similar manner, the cycle time of assembly modules can be determined to meet trunk line by using the redundant-stations in modules.

## VI. Scheduling Approach For Robustness

### A. Scheduling Rules

Long-term equipment failure is inevitable in RAS. Our method for increasing robustness of RAS is to find the "substitute device" in other assembly modules and to change workflow through the special tracks. So there two kinds of parts are waiting for assembly, and how to arrange them is a key problem. In a network diagram, when two concurrent activities can't start simultaneity any more by reason of resource conflict, one of them has to start after the completion of the other. We must determine the precedence relation between them appropriately and coordinate the conflict. The aim is to minimize the increased value of execution time of the project. As shown in Fig.6, A and B are two concurrent activities, because they need to utilize the same resource, B is postponed after A, i.e. the precedence relation is A→B. The corresponding increased value of execution time of project is $\Delta T_{AB}$, and the expression is as follows:

$$\Delta T_{AB} = EF_A + t_B - LF_B \tag{9}$$

Because $LF_B - t_B = LS_B$ , the following formula is obtained:

$$\Delta T_{AB} = EF_A - LS_B \tag{10}$$

If $\Delta T_{AB} \le 0$ , it shows that the execution time of the project doesn't become longer with the precedence relation A→B.

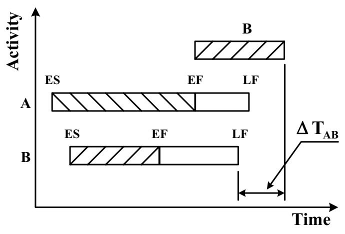


Fig.6 The increased value of execution time

To determine the precedence relation of two activities, $\Delta T_{AB}$ and $\Delta T_{BA}$ are calculated by the formula (10) firstly, and then select the relation with the minimum increased value. The concrete scheduling rules are as follows:

if $\Delta T_{AB} < \Delta T_{BA}$ , then A→B;

else if $\Delta T_{AB} > \Delta T_{BA}$ , then B→A;

else $\Delta T_{AB} = \Delta T_{BA}$ , then randomly arrange.

### *B. Experimental Case*

As shown in Fig.7, trunk line and two assembly modules are in the original network diagram. An imaginary line represents the interfaces between trunk line and the corresponding modules.

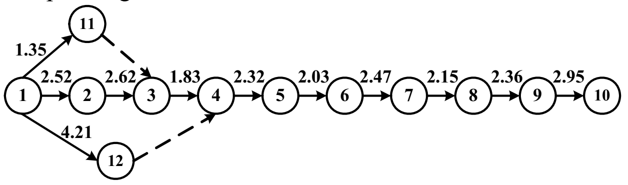


Fig.7 Original network diagram

The activity (1, 11) is the assembly module for middle-shaft of gear-box product, and (1, 12) is the output-shaft module. If the pressure machine in (1, 11), noted as A, is broken, the substitute can be find in (1, 12) noted as B. In order to resist the disturbance, we must reconfigure the assembly system and transfer the activity A to B. Resource conflict will occur between the two concurrent activities, and precedence relation need to be determined in order to minimize the increased value of production time. Using CPM/PERT, the evaluation of $\Delta T_{AB}$ and $\Delta T_{BA}$ is given in Table I.

TABLE I
EVALUATION OF $\Delta T_{AB}$ AND $\Delta T_{BA}$

| Item | Value |
|---|---|
| $EF_A$ | 1.35 |
| $LS_B$ | 2.62 |
| $EF_B$ | 4.21 |
| $LS_A$ | 3.79 |
| $\Delta T_{AB}$ | -1.27 |
| $\Delta T_{BA}$ | 0.42 |

From the information of Table I, the precedence relation is determined as A→B. The network diagram of the new configuration is shown as Fig.8.

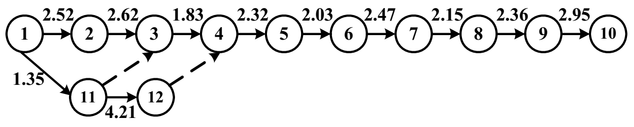


Fig.8 Network diagram of new configuration

It can be seen that the pressure machine in (1, 12) is charged with two operations including the assembly of middle shaft and output shaft.

## VII. CONCLUSIONS

In this paper, the common organizational structure for RAS is proposed based on the modelling of a product family. This common structure has many predominant characters such as capacity flexibility, process robustness, and scalability flexibility and so on. The modularization of RAS increases the responsibility to market and facilitate the reconfiguration process as well. Using CPM/PERT, a scalable algorithm and a scheduling approach are given to realize the scalability and robustness of RAS respectively. The further study of the common structure is to determine the size of the flexible buffer, and the layout of the special tracks.

## ACKNOWLEDGEMENT

The authors would like to thank all member companies for their supports. The authors are also grateful to WCICA 2006 for the communication opportunity.

## REFERENCES

[1] M.G. Mehrabi, A.G. Ulsoy, and Y. Koren, "Reconfigurable Manufacturing Systmes and Their Enabling Technologies," International Journal of Manufacturing Technology and Management, vol.1, no.1, 2000, pp. 113-130.

[2] Y. Koren, U. Heisel, and F. Jovane, "Reconfigurable Manufacturing Systems," A Keynote paper. CIRP Annals. vol.48, no.2, 1999, pp. 527-540.

[3] D. Popa, B.H. Kang, J. Sin and J. Zou, "Reconfigurable Micro-Assembly System for Photonics Applications," Proceeding of the 2002 IEEE International Conference on Robotics & Automation, vol.2, 2002, pp. 1495-1500.

[4] Z. Kong and D. Ceglarek, "Rapid Deployment of Reconfigurable Assembly Fixtures using Workspace Synthesis and Visibility Analysis," Annals of the CIRP, vol. 52, no.1, 2003, pp. 13-16.

[5] A. McKay, F.J. Erens, and M.S. Bloor, "Relating product definition and product variety," Research in Engineering Design, vol. 8, no. 2, 1996, pp. 63–80.

[6] P. De Lit, T. L'Eglise and A. Delchambre, "Functional Entities: A Concept to Support Product Family and Assembly System Design," Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning, 2001, pp. 160-165.

[7] K.T. Ulrich and S.D. Eppinger, Product Design and Development, Irwin McGraw Hill Publications, Second Edition, 2000, pp. 14-18.

[8] S. Bussmann and J. Sieverding, "Holonic control of an engine assembly plant: an industrial evaluation, Systems," Systems, Man, and Cybernetics, 2001 IEEE International Conference, vol.1, 2001, pp. 169-174.

[9] Z. Zhu and R. Heady, A. Delchambre, " A Simplified method of evaluating PERT/CPM network parameters," IEEE Transactions on Engineering Management, vol.41, no.4, 1994, pp. 426-430.