2003 Deliverable Report:
Forecasting methodologies for multidimensional aggregated demands

Task 879.001: Intelligent Demand Aggregation and Forecast Solutions
Project 879: Demand Data Mining and Planning in Semiconductor Manufacturing Networks

Task Leader: Argon Chen
Co-PI's: Ruey-Shan Guo and Shi-Chung Chang
Students: Tonny Huang, Kyle Yang and Janet Tzeng

1. Abstract and Summary

In 2002, to cope with demand uncertainty, we have defined and proposed optimum demand planning hierarchy (DPH) to best support different granularities of planning activities. (see 2002 deliverable report). This year, the proposed DPH is extended from one-product DPH to hierarchical-product DPH. In addition, the DPH concept has been realized in a software system prototype, referred to as DPH Planner. In the DPH planner, the optimum DPH with the least demand fluctuation can be automatically created. Moreover, users can define their current demand planning practice in the DPH planner. With the benchmark of the optimum DPH, a what-if interface is also provided to allow users adjust their DP practice.

2. Technical Results

2.1 Hierarchical Products

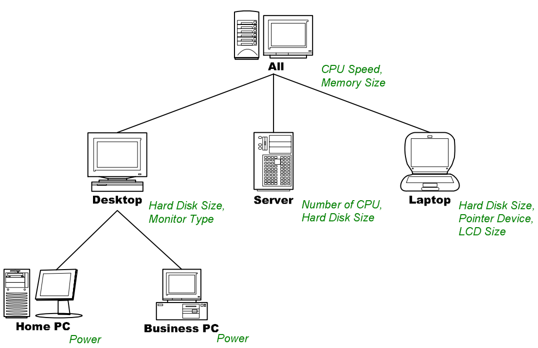


Fig. 1 Product Hierarchy Example

For the hierarchical DPH, the product dimension has been enhanced to be a hierarchical product dimension. The product dimension can be classified into two categories: product group and product attribute. Product groups form a hierarchical tree. The highest aggregated product group in the product hierarchy is called "root group" and the group name must be "All" (as shown in Figure 1). Unlike the product groups, the product attributes are independent of one another. We further classify product groups in the tree structure into two types: (1) internal node (or internal group) and (2) leaf node (or leaf group) [6]. Internal groups are the groups that can be further split and leaf groups are the product groups that cannot be further split. In Figure 1, "All" and "Desktop" are internal groups. "Home PC", "Business PC", "Server", and "Laptop" are leaf groups. A higher-level internal product group is also called a "parent product group" (or in short "parent group") to its lower-level groups that are called "child product groups" (or in short "child group"). For the example in Fig. 1, "Desktop" is the parent group of its child groups: "Home PC" and "Business PC". Every non-root group has one and only one parent group. Every internal group has more than or equal to two child groups. A product attribute represents a feature of a product group. In this research, we assume only common features will be considered for a product group. In other words, the attributes defined in a product group must exits in all products of the group. A product hierarchy includes more than one product groups so we have to consider different attributes of different product groups in the hierarchical-product DPH. We classify product attributes into two types: (1) private attribute: the attribute that is defined in a leaf group; and (2) common attribute: the attribute that be defined in an internal group. Common attributes are inheritable. A child group will inherit the attributes of its parent group. Take Figure 1.4 as an example, both "Home PC" and "Business PC" are sold with monitors so we can define the attribute "Monitor Type" in their parent group "Desktop". In other words, a common attribute defined in a product group must exist in all of its child groups.

2.2 DPH Representation System for Hierarchical Products

For a hierarchical product dimension, there are two components in its dimensional view: (1) product attribute, and (2) product group. For product attributes, we adopt the symbolic representation for "views with attributes", as introduced in section 1.2. We use capital letters to specify the dimension name and the subscript is used to express the dimensional view. To express the combination of product attributes, the symbols " • " and " × " denote the hierarchical and combinational relationships, respectively. To express the corresponding attributes in a product group, we use parentheses "( )" to group its corresponding attributes. For example, $\text{Product}_{\text{Desktop (CPU Speed} \times \text{Hard Disk Size)}}$ represents a product dimensional view of product group "Desktop" with the combination of product attributes "CPU Speed" and "Hard Disk Size". There may be more than one product group being considered for the product dimension. Symbol "+" is placed between two product group expressions to represent a complete product dimensional view. For example, "$\text{Product}_{\text{Desktop (CPU Speed} \times \text{Hard Disk Size)} + \text{Server (CPU Speed)} + \text{Laptop (CPU Speed)}}$" represents that current product differentiation is: "Desktop", "Server", and "Laptop" with their corresponding attributes in the parentheses.

A multidimensional demand view is a combination of multiple dimensional views. Symbol "×" is used to concurrently represent these dimensional views. For example, suppose three dimensions are analyzed: "TIME" (with three hierarchical levels: "Quarter", "Month", and "Day") and "CUSTOMER" (with two hierarchical levels: "Region" and "City"), and "PRODUCT" with a product hierarchy shown in Figure 2.3. The expression "$PRODUCT_{All}$ × $CUSTOMER_{Region}$ × $TIME_{Quarter\cdot Month}$" represents that we are looking at the monthly demand for aggregated products of different regions. Fig. 2. shows another example of a multidimensional demand view with multiple product groups. The expression of multiple product groups depends on the product differentiation.

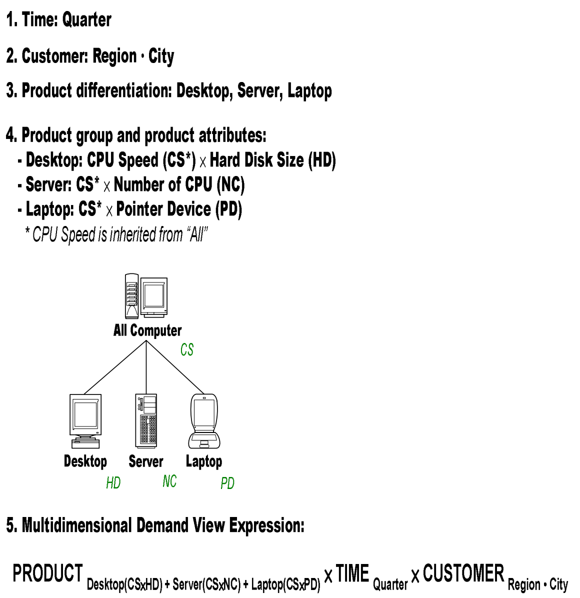


Fig. 2 Example of Multidimensional Demand Viewn

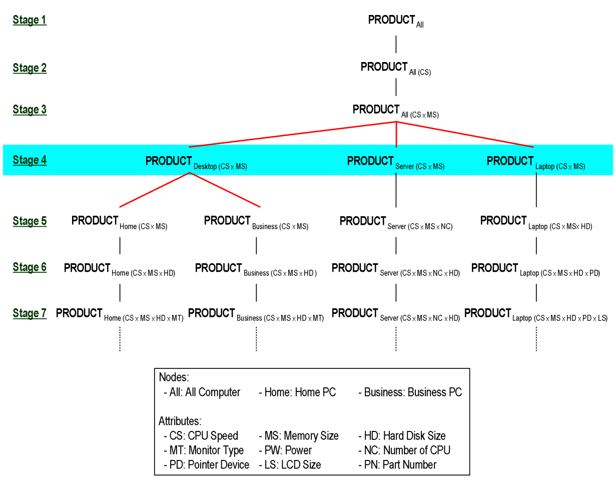


Fig. 3 Demand Planning Hierarchy Representation

We have defined the expression of a multidimensional demand view for products with a hierarchical structure. As the expression defined above, product dimension is the lead dimension of the multidimensional demand view expression and is the main focus of our research. In this section, we consider only on the product dimension. As we mentioned, a DPP is said to be a sequence of multidimensional demand views. Based on the expression defined previously and to emphasize hierarchical relations among product groups, we propose a representation system shown in Fig. 3.

2.3 Evolving Rules for Hierarchical Product Dimension

A product group inherits the attributes of its parent group. We assume an inherited common attribute must be considered by all child product groups at the same time. That is, a common attribute must be chosen by all the child groups to aggregate or disaggregate. Fig. 4 shows an invalid and a valid example of disaggregating a common attribute.

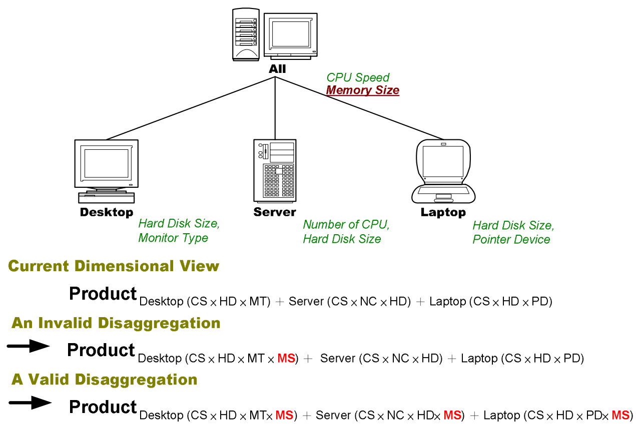


Fig. 4 Evolving Rules for Common Attributes

In the example above, the attribute "Memory Size" is a common attribute at the highest aggregated level and is chosen to disaggregate the demand. Since it is a common attribute of product groups "Desktop", "Server", and "Laptop", this attribute must be used to disaggregate by all of the child groups together. Similarly, aggregation of an inherited common attribute should also be adopted by all of the product groups at the same time.

We also propose two disaggregation priority options for the product attributes: (1) prior to product group splitting and (2) prior to lower-level attributes. The "prior to product group splitting" disaggregation priority forces disaggregation through all the common attributes before splitting the product group. Fig. 5(a) shows an example where the attribute "Memory Size" has this priority. We can't split the product group "All" before disaggregating the demand by this attribute. Similarly, this priority can be considered as an aggregation priority Fig. 5(b) shows a valid and an invalid example where the attribute "Memory Size" has this priority. We must merge the product groups "Desktop", "Server", and "Laptop" before aggregating the demand by the attribute "Memory Size".

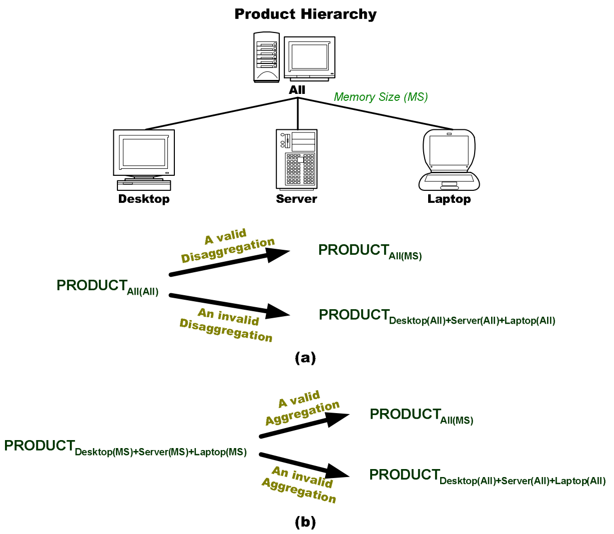


Fig. 5 The "Prior to Product Group" Options

The "prior to lower-level attributes" disaggregation priority forces disaggregation through all the higher-level attributes before disaggregation through the lower-level attributes. In Fig. 6, the product attribute "Memory Size" is the common attribute of product groups "Desktop", "Server", and "Laptop" and thus is an attribute at the higher level than attribute "Monitor Type", "Number of CPU", and "Pointer Device". When the "Memory Size" has the priority, the disaggregation by "Memory Size" must be performed before all other lower-level attributes. Fig. 6(a) shows a valid and an invalid disaggregation. Similarly, this priority can be considered as an aggregation priority. Fig. 6(b) shows a valid and an invalid example where the attribute "Memory Size" has this priority and aggregation must be done through all lower-level attributes first before aggregating by "Memory Size".

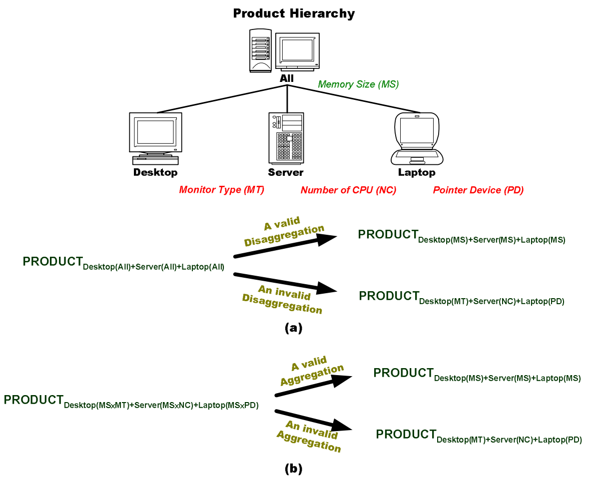


Fig. 6 The "Prior to Lower-level Attribute" Option

2.4 DPH What-if Analysis

2.4.2 User Defined Hierarchy (UDH)

The sequence of multidimensional demand views considered by decision makers may not be very detailed. These views form a DPH that is called a user defined hierarchy (UDH). However, these multidimensional demand views are probably violating the stepwise evolving rule or the evolving rules for product differentiation. That is, a UDH is probably an incomplete DPH. The DPH analyzer allows decision makers to define an incomplete user defined DPH. Decision makers can sequentially define each stage of their current demand planning practice into the UDH by the provided interface. A UDH represents the current demand planning strategy. The purpose of defining a UDH is to compare the current corporate demand planning strategy to the suggested demand planning strategy solved by the DPH optimization. However, the comparison is not easy if the UDH is not a complete DPH. To make them comparable, the first task of DPH planner is to transform the incomplete UDH to a complete one. The procedure is shown in Fig. 7

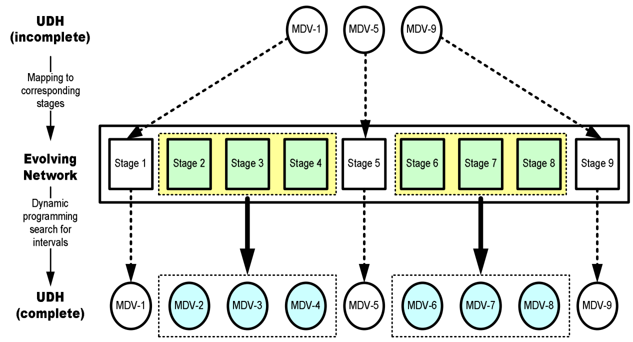


Fig. 7 Generating a Complete UDH

In Fig. 7, a node represents a multidimensional demand view. The DPH analyzer first maps every user defined multidimensional demand views to their corresponding stages in the system-defined evolving network. If the UDH is not a complete DPH, there must have some non-mapped stages. For the non-mapped stages, DPH analyzer adopts dynamic programming search to find the absent multidimensional demand views in the non-mapped stages to form the shortest paths in between the mapped stage. Thus, the incomplete UDH become a complete one.

2.4.2 UDH What-if Analysis

To compare the UDH and the solved optimal DPH, a what-if analysis tool called UDH tuner is designed for such a requirement. Fig. 8 shows the user interface of the UDH tuner. UDH tuner uses line chart to show the difference between the UDH and the optimal DPH. The line which represents the UDH is always the upper line in the line chart because the solved DPH is the optimal solution. The objective is to allow the decision makers to make these two lines as close as possible. If decision makers apply the optimal DPH as their UDH, the lines will overlap with each other. The line chart is very helpful to diagnose the UDH which represents the current demand planning strategy. Taking Fig. 8 for example, the upper line is significantly higher than the lower line. It means that the current demand planning practice has room to improve. However, it is usually unacceptable for a company to fundamentally change their demand planning practice. DPH analyzer allows the real-time tuning of the user defined hierarchy by aggregating or disaggregation a user defined multidimensional demand views. Taking Fig. 8 for example, the multidimensional demand view of the fifth stage may need to be changed because the distance between the lines becomes significantly large starting this stage. If a multidimensional demand view is changed, all components in the interface will be updated. Therefore, decision makers can dynamically tune the UDH for what-if analysis and to fine the most acceptable hierarchy for their need.

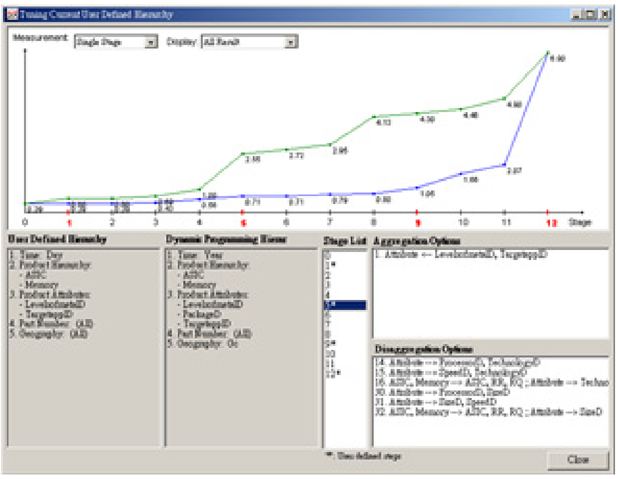


Fig. 8 UDH Tuner User Interface

2.5 Semiconductor Demand Case Study

The case study dataset consists of three dimensions: "Time", "Customer", and "Product". Dimension "Time" and "Customer" are dimensions with hierarchical levels and product dimension is a hierarchical product dimension. There are five dimensional views in the dimension "Time": "Year", "Quarter", "Month", "Week", and "Day". Any feasible hierarchical structure of these dimensional views can be used for this dimension. For example, the hierarchical levels: "Year", "Week", and "Day" is a hierarchical structure. However, the hierarchical levels: "Year", "Quarter", "Week", and "Day" is an infeasible structure since "Quarter" does not directly consist of weeks. For dimension "Customer", an individual customer account must belong to one geographical code. We define three dimensional views in the dimension "Customer": "GC" stands for individual customers, "GG" stands for higher level geographical codes and "Global" (or "All") stands for the union of all customers. There are 81 customers and 4 geographical codes considered. Two main groups of products are considered in dimension "Product": "ASIC" and "Memory". The product group "Memory" can be further split to two child groups: "RR" and "RQ". The product hierarchy is show in Fig. 9. There are four common attribute and four private attribute defined in the product hierarchy above. The attributes: "Technology", "Levels of Metal", and "Size" are private attributes that used to characterize the product group "ASIC". The attributes "Target Appl.", "Processor", and "Speed" are common attributes that used to characterize the product group "Memory" and are inherited by product groups: "RR" and "RQ".

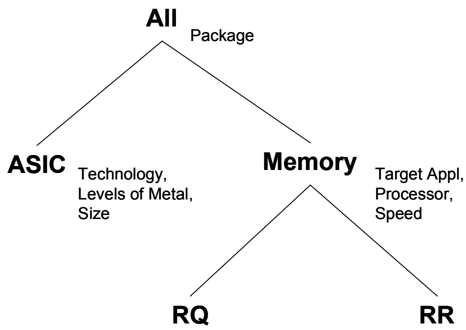


Fig. 9 Semiconductor Product Hierarchy

After obtaining the optimal hierarchical-product DPH by dynamic programming (see 2002 deliverable report), the line chart that comparing the current user defined hierarchy and the optimal DPH is shown in Fig. 10.

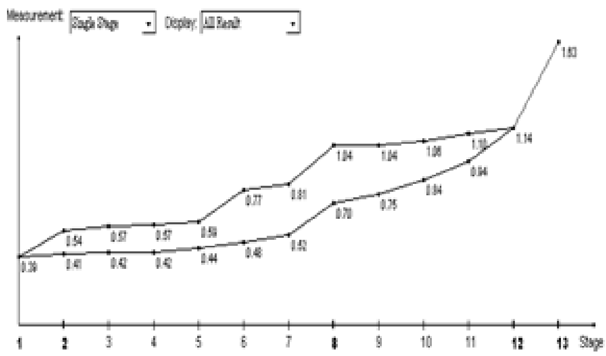
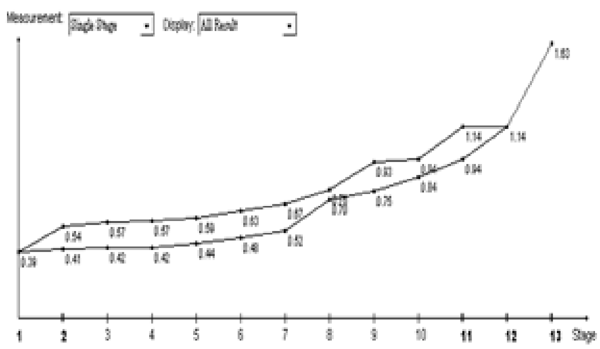


Fig. 10 Comparison b/w UDH and Optimal DPH

In the UDH tuner, we can adjust the multidimensional demand views we defined to see the possible chances of improvement. For the example above, we select the 8th stage for tuning. The line chart of the tuned UDH is shown in Fig. 11 and the improved hierarchy structure is shown in Fig. 12.



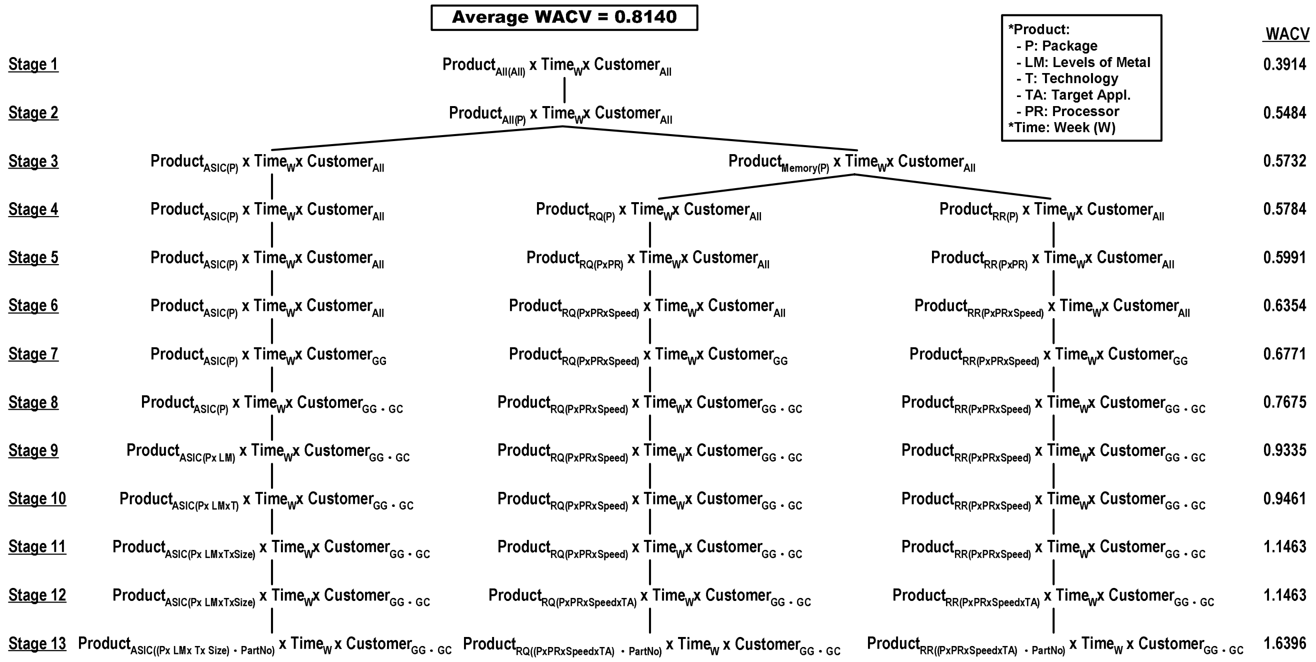


Fig. 12 User Defined Hierarchy after What-if Tunning