Dependent Demand -
a Business Pattern for Balancing Supply and Demand

Version 7.0 10-15-97

Introduction:

This is the first pattern in a process that will develop a pattern language for supply chain management.

The concept of dependent demand was the main intellectual contribution of the software movement known as Material Requirements Planning or MRP. It is well-documented [1] and well-known within manufacturing software circles [2].

The motivation for writing up dependent demand as a pattern is that a) the applicability of the idea has broadened significantly since its conception; b) these broader uses are not well-documented; and c) some opportunities for using dependent demand are being missed (for example, duplicate buffer stocks kept by both companies involved in consumer-supplier relationships).

Traditional MRP dealt with dependent demands for materials (only) within a single manufacturing plant.

More recent software, and this paper, extend the pattern to deal with all resource requirements including people, machines and tools.

Moreover, supply chain time compression has emerged as a major focus of business competition. Given some proven ability to help compress time, Supply Chain Management software has become a billion-dollar software market category. These phenomena motivate the extension of dependent demand networks to cover multiple companies and the transportation between them.

Type of Pattern:

Dependent Demand is not a design pattern. It could be (and has been) implemented many different ways. It is a pattern of relationships between business events and the generalized (abstract) technique for managing these relationships.

Definitions:

Independent Demand is any demand that has no cause within the business-system context (although it may have a cause in a larger context). For example, to an automobile dealer, a customer order for a new car is an independent demand (although to the customer, it might have been dependent on the unreliability of the customer’s old car).

Dependent Demand is any demand that is caused by an independent demand, or is necessary to the satisfaction of the independent demand. For example, an independent demand for a new car causes dependent demands for all of the components which make up the car.

Pattern Title: Dependent Demand

Problem: Balancing supply and demand in complex multi-stage processes.

Context: Supply chain management for commercial products.

Dependent demand is by far the most common type of demand in distributed multi-stage networks of manufacturing and distribution: for example, the automobile industry, where cars are purchased in dealer showrooms, delivered to the showrooms by trucks from factories where they are manufactured on assembly lines, which are fed by several tiers of parts manufacturers (e.g headlights, seats, wheels, engines, pollution control devices, etc. etc.), which are in turn fed by basic material industries (steel, glass, rubber, etc.)

The automobile networks are world-wide and well-known, but there are analogous networks for most commercial products: for example, computers, where time compression has become a critical success factor.

Forces:

- There are very few truly independent demands in supply chains, and most other demands are dependent on those few independent demands.

- However, it can be very difficult to manage the dependencies, especially when they change. Especially if the business administrative systems do not recognize the dependencies in the first place. (For example, manually-entered Purchase Orders for dependent components which are disconnected from the independent demand they were purchased for: if the independent demand changes, the Purchase Order will not automatically be changed to correspond.)

- Imbalances in supply and demand result in unsatisfied demands coupled with wasted supplies and efforts.

- Complex products with many stages and components tend to have complex supply chains.

- Unpredictable or lumpy demands make it difficult to balance supply and demand. Unpredictable demands for complex products cause ripples of unpredictability that become chaotic.

- When product categories undergo rapid change, it is unwise to simply restock the product that was purchased, because the restocked item may have no further demand, while some new item may be in demand.

  For example: the recurrent over-and-under-supplies in the PC industry, where the model in demand is almost always not the one the dealer has in stock. Computer makers such as Dell are moving to a dependent demand pattern, where the computers are built-to-order and components can be obtained on short notice (in Dell's case, 15 minutes). (Since Dell started this practice, its competitors are climbing on the bandwagon to the extent that the pattern has acquired an acronym: BTO.)

Applicability:

- For dependent demands to work as a management technique, it must be possible to trace from independent to dependent demand in some predictable way.
- Moreover, the output from one stage of production must be the input to a subsequent stage, again in a fairly predictable way.
- On the other hand, if all stages of production and distribution are totally repetitive and predictable, while dependent demand still operates as a pattern, it is overkill as a management technique. Simpler techniques might be better, such as **Kanban** systems as used by the Toyota Production System [5] (although some Kanban techniques are really manual implementations of Dependent Demand). Kanbans are cards used to signal requirements between consumer and supplier processes.
- In some product categories where independent demands are predictable, dependent demands may not be predictable because of Murphy's law, among others. (For example, in food processing, supplies may be unpredictable and so may be dependent demands.) Other demands that are dependent on these variable ones need to be managed by dependent demand techniques.
- On the other hand, dependent demand is difficult to use in contexts such as research projects where the process is too unpredictable.
- Dependent demand is most useful in demand-driven contexts (that is, where demand can be known or somewhat reliably forecast).
- In supply-driven contexts such as agriculture, mining or such industries as paper mills or petroleum refineries, dependent demand is either irrelevant or useful only in broad aggregates.

Solution:

For each independent demand:
- determine all of the input components and production resources required to satisfy it;
- model these as dependent demands;
- match them with planned supplies;
- repeat this process recursively until all dependent demands have reserved supplies;
- and maintain the resulting network of demands and supplies in persistent relationships until the independent demands are fulfilled and the need for a history of the process is over.

Starting on the next page is a sketch of a feasible object-oriented implementation of the solution -- although it should be understood that most existing implementations are not object-oriented (and some are not even computerized).

Core Abstractions:

- Activity
  - ManufacturingActivity
  - ProcurementActivity
  - TransportationActivity
- Demand
  - IndependentDemand
  - DependentDemand
- Resource
  - MaterialResource
    - Product
  - CapacityResource
- Role
  - Consumer
  - Supplier
- Specification (a pattern presented at Plop '97 [7])
- Supply
  - OnHandSupply
  - PlannedSupply

There must also exist a **Knowledge Level** (from Martin Fowler [3]) which guides the explosion of dependent demand. (I am warping the meaning of Knowledge level a little bit here, but I mean an information service which may be separate and independent from the Dependent Demand network. This information service can take many forms.)

The Knowledge Level is responsible for knowing the *paths* to find such information as:

- the component parts for a demanded Product;
- the on-hand inventories of demanded Products;
- the process (sequence of Activities) for manufacturing or purchasing demanded Products if on-hand inventory is insufficient;
- the suppliers of demanded Products and CapacityResources for ManufacturingActivities;
- the schedules for CapacityResources.

In most MRP systems, this information is stored in:

- **Bills of Material**, which determine the component parts for products;
- **Routings**, which determine the manufacturing operations and their CapacityResource requirements;
- **Inventory**; and
- **Purchasing** systems.

In more modern systems, Bills and Routings (and also sometimes Inventory and (less often) Procurement) are combined into the same structure sometimes called **Bill of Manufacture** or **Process Model**.

The Bill of Material function could also be filled by a rules-based system sometimes called a **Configurator**. (This is the norm in the computer hardware business.)

Dependent Demand Explosion:

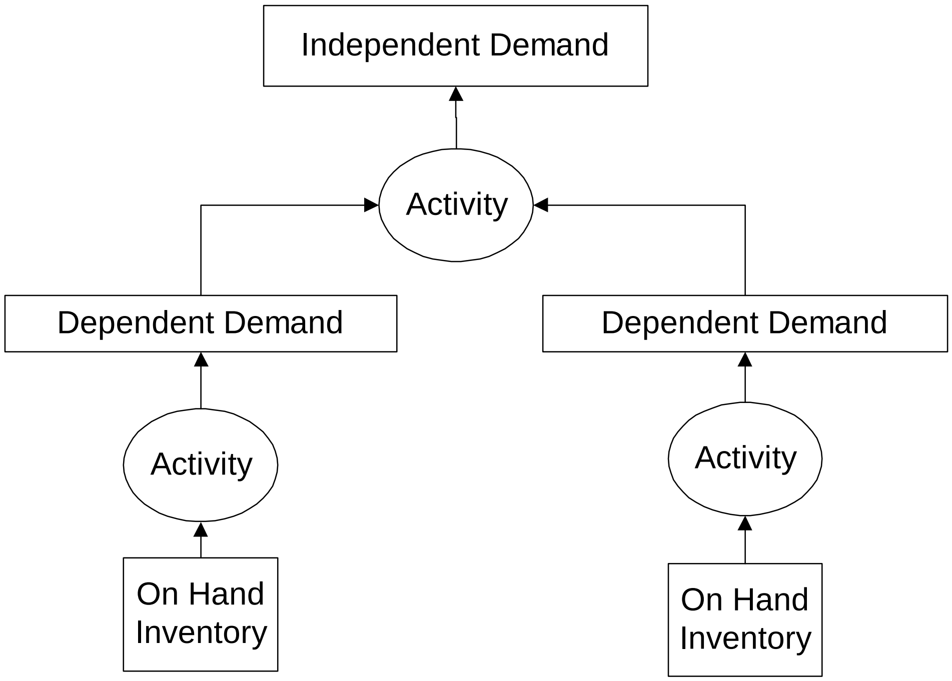
The explosion proceeds backward from the end of the process, which is the delivery of the IndependentDemand.

The procedure is recursive and generates a tree structure; the recursion terminates when each branch's demands are satisfied by onhand inventory (for Products) or available schedule time (for CapacityResources) (or exceptions are raised because there is no more knowledge available).

The following procedure focuses on IndependentDemands for Products; Services are similar in concept but may differ in detail.

1. A Demand for a Product asks the Knowledge Level to find a Supplier. This Supplier could be internal to the demanding company, for example another department in the same manufacturing plant, or it could be external: a different company. The Knowledge Level could contain a collection of Supplier candidates and pick the most appropriate one based on the Demand's Specifications.
2. *Standard MRP systems usually do **Netting** (a sub-pattern or idiom) at this point in the explosion procedure. However, fast-and-fluid environments would be better off to skip netting, because inventory updates may be too slow to keep up with the changes, and they usually try to keep nothing on hand to net anyway.*

   Thumbnail sketch of **Netting**: If the Supplier has sufficient Supply available (on hand or planned) to satisfy the Demand, then the available Supply is allocated, and the Demand is satisfied (this branch of the tree terminates).
   If the available Supply only partly satisfies the Demand, then the explosion continues based on the remainder of the Demand.
3. A new PlannedSupply is scheduled to satisfy the unfilled Demand.
4. For the new PlannedSupply, the Supplier will schedule an Activity to manufacture or procure the Product required by the Demand.
5. If the Activity is a ManufacturingActivity, then it will generate DependentDemands for the Resources required to manufacture the Product: for example, DependentDemands for component Products, a DependentDemand for the IndependentDemand Product at its previous stage of development, and DependentDemands for CapacityResources (machine and labor time of the required types).
6. Each of the DependentDemands for component Products will recursively ask the Knowledge Level to find a Supplier to manufacture or procure their requirements.
7. (Probably) more DependentDemands will result, which will of course go through the same procedure.


Independent Demand
Activity
Dependent Demand
Dependent Demand
Activity
Activity
On Hand Inventory
On Hand Inventory

Examples:

Simple example:

To make bread, you will need 4 cups of flour and 1 package of yeast and a teaspoon of salt, and then you need to schedule time to mix, rise, knead and bake in the oven. If you don't have the flour and yeast, you need to go to the store to buy them.

In this example, bread is the Independent Demand. The recipe book is the Knowledge Level. Flour, yeast, salt, mixing time, rising time, kneading time, and baking time in the oven are Dependent Demands.

More complex example:

The following example (on the next 2 pages) might be a trace from a Dependent Demand Explosion computer program for an automobile manufacturing company (in fact, it is based on a series of debug print messages from a Smalltalk explosion program).

The Knowledge Layer in this example is a Bill of Manufacture, which contains all of the required knowledge in one structure.

Besides Dependent Demand, there are a few sub-patterns or idioms in this example which are invoked when planning supplies:

- **Lot Size Rules** govern what quantity should be planned.
- **Periodic Lot Size Rules** (like **Week's Supply**) mean that the explosion procedure must table the demands for the affected component until all registered demands have been collected.
- **Safety Stock** strategies try to keep some of a component on hand at all times; when the unallocated onHand quantity drops below the safety stock level, more is ordered.
- **Gang Daily Deliveries** (from the same vendor), which humans usually do manually.

These sub-patterns also demonstrate that Dependent Demand is rarely used in any pure manner. See **Related Patterns** below for more of the same.

```
---------------Start trace--------------------

Automobile Indented Bill of Manufacture in explosion sequence:
    ManufacturingActivity: Assemble
        Dependents:
            1 hour on Final Assembly Line
            1 Chassis (delivered) - Lot Size Rule: Min 20
                TransportationActivity: Deliver
                    Dependents:
                        5 hours from Trucking Company
                        1 Chassis (finished)
           1 Engine - Lot Size Rule: Week's Supply
               ManufacturingActivity: Assemble
                   Dependents:
                       .5 hr on Engine Assemble Line
                       1 Block
                           ManufacturingActivity: Machine
                               Dependents:
                                   2 hrs on Flexible Machining Center
                                   1 Engine Block Core Strategy: Safety Stock 10
                       4 Pistons - Strategy: Gang daily deliveries for supplier
                           TransportationActivity: Deliver
                               Dependent:
                                  1 hr from Engine Parts Company
                       4 Connecting Rods - Strategy: Gang daily deliveries for supplier
                           TransportationActivity: Deliver
                               Dependent:
                                  1 hr from Engine Parts Company
                       1 Crankshaft - Strategy: Gang daily deliveries for supplier
                           TransportationActivity: Deliver
                               Dependent:
                                  1 hr from Engine Parts Company

On hand inventory:
   1 Automobile
  20 Chassis (finished) at Chassis Company
  20 Engine Block Cores

Orders:
  Order Number 101: 2 Automobiles due 6-25-97
  Order Number 102: 5 Automobiles due 6-26-97
```

```
---------------Explosion trace:--------------------

Order 101:
  Demand: 2 Automobiles
    Allocate 1 onHand Automobile
    Plan Activity Assemble to supply remaining demand of 1 Automobile
      Dependent demand: 1 hr on Final Assembly Line
        Reserved 6-24-97 8AM-9AM (negotiated with Scheduler)
      Dependent demand: 1 Chassis delivered before 6-24-97 8AM
        Demand for 20 Chassis (delivered) placed with Trucking Company
       includes dependent demand of 1
          plus independent demand of 19 because of Lot Size Rule.
      Dependent demand: 1 Engine
        tabled because of Week's Supply Lot Size Rule.

Order 102:
  Demand: 5 Automobiles
    Plan Activity Assemble to supply 5 Automobiles
      Dependent demand: 5 hrs on Final Assembly Line
        Reserved 6-24-97 9AM-2PM (negotiated with Scheduler)
      Dependent demand: 5 Chassis delivered
     Allocated from planned unallocated supply
          of 19 Chassis due 6-24-97 8 AM.
      Dependent demand: 5 Engines
        Demand for 6 Engines
          because of Week's Supply Lot Size Rule
            1 allocated to Order 101
            5 allocated to Order 102
          Plan Activity Assemble to supply 6 Engines
            Dependent demand:  2.5 hrs on Engine Assembly Line
              Reserved 6-24-97 2AM-4:30AM (negotiated with Scheduler)
            Dependent demand: 5 Blocks
              Planned Activity Machine
                Dependent demand: 10 hrs Flexible Machining Center
                  Reserved 6-23-97 8AM-6PM
                Dependent demand: 5 Block Cores
                  Allocated 5 onHand Block Cores
                    did reduce onHand below Safety Stock.
            Dependent demand: 20 Pistons
              tabled to gang Engine Parts demands.
            Dependent demand: 20 Connecting Rods
              tabled to gang Engine Parts demands.
            Dependent demand: 5 Crankshafts
              tabled to gang Engine Parts demands.

Delivery scheduled from Engine Parts Company
  due before 6-23-97 8AM including:
    20 Pistons
    20 Connecting Rods
     5 Crankshafts.

---------------End trace--------------------
```

Known uses:

Every MRP and MRPII system has the dependent demand pattern as its heart and soul. So do most so-called Enterprise Resource Planning or ERP systems.

The extended all-resources total supply chain dependent demand pattern is used in one form or another by i2, Manugistics, Red Pepper, and Berclain, among others [4].

Jeff Sutherland reports via email that the Mayo Clinic does a manual dependent demand explosion when a patient flies in for treatment.

Resulting context:

There will be a persistent directed graph of activity nodes connected by supply-demand links that lead to independent demands.

Changes anywhere on this network can be propagated to all affected activities.

Execution of activities will occur on top of this persistent structure: the dependent demand network is also the work flow network. In some cases, even payment for component parts is computed by dependent demand explosion, eliminating purchase orders and invoices (Ford and Saturn are doing this).

Problems can be raised to the highest level of demand that they affect by following the arrows in the graph.

When the independent demand has been satisfied, the history of events supplies the structure for performance measurement and tracing back to the source if problems occur when the end product is being used. (For example, when tainted food was found on grocery shelves, the manufacturer used the lot number on the package to trace through the dependent demand records to find the sources of the package and all its ingredients.)

Related Patterns:

Dependent Demand is not the only pattern involved in Supply Chain Management. In fact, one could write a Supply Chain Pattern Language, with Dependent Demand as the center. Here are some of the other patterns, idioms and techniques that may be included in such a pattern language:

In most MRP systems, the dependent demand explosion assumes infinite capacity for capacity-constrained resources (e.g. machines and people). Such an assumption, needless to say, cannot hold. So subsequent patterns or techniques are required to schedule capacity-constrained resources. One such technique is **Constraint Logic Programming** [6], (or just Constraints) which can synchronize both the capacity and material resource schedules (once the dependent demand network has been generated).

Even with a strong constraint solver, however, usually some resource conflicts cannot be resolved without appealing to human judgment. Thus a good interactive **Trouble-Shooting** component is required.

Almost always, time and other constraints prevent using pure Dependent Demand in a supply chain. For example, although Dell tries to keep no inventory on hand (relying on pure independent demand), the same is not true of its suppliers, who warehouse components locally.

So some of the demands in the supply chain need **Forecasting** (which of course is an imprecise art, and the source for some other patterns.) But for the purpose of generating dependent demands, forecast demand acts as independent demand as well as customer orders.

Moreover, many variations on the explosion pattern outlined above are caused by local efficiency requirements, such as grouping and **Sequencing** dependent demands to minimize changeovers on capacity-constrained resources.

More subordinate patterns and idioms (used to support dependent demand) include:

- **Netting**, which allocates available supplies before scheduling new supply activities;
- **Backward and Forward Scheduling;**
- **Pegging**, used by the Trouble-Shooting component to find demands affected by resource and schedule problems; traverses the dependent demand network;
- **Lot Tracing,** finding the source lots when an end product is bad, or finding the destinations of bad ingredient lots; traversing the dependent demand network in different directions. (Lot is roughly equivalent to a batch of product with a persistent identification key.)

And, of course, the material control idioms cited in the **Complex Example** above:

- **Lot Size Rules** govern what quantity should be planned.
- **Periodic Lot Size Rules** (like **Week's Supply**) mean that the explosion procedure must table the demands for the affected component until all registered demands have been collected.
- **Safety Stock** strategies try to keep some of a component on hand at all times; when the unallocated onHand quantity drops below the safety stock level, more is ordered.
- **Gang Daily Deliveries** groups deliveries from the same supplier for efficiency.

Summary:

The essence of dependent demand as a pattern is that a dependent demand is dependent in quantity, quality and timing on its related independent demand.

To the extent that the supply chain can conform to those rules, the independent demands will be satisfied with minimal wasted supplies and efforts.

Moreover, the dependent demand network provides a context for scheduling, operations, cash flows, and historical analyses such as source traceability.

Acknowledgements:

Thanks for the feedback to my PLoP shepherd, Gerard Meszaros; the members of the Chicago Patterns Group, where this pattern was introduced; the discussion participants from the Business Patterns mail list, Ralph Johnson, Stephen McHenry, Thomas Mercer Hursh, Martin Rojo, Paul J. Ste. Marie, and Shalom Reich; and the members of the PLoP ’97 workshop: Dirk Riehle, F. Balaguer, Wolf Siberski, Eric Evans, Martin Fowler, and Joseph Yoder.
(Needless to say, none of them are not responsible for anything you dislike about the current paper.)

References:

MRP was developed in the 1960's and '70's mostly in the US and Canada. It is the standard model for manufacturing software in the North America. MRP is not standard in continental Europe or Asia - although it is used in both regions.

MRP and dependent demand are described in numerous books including this classic:

[1] Orlicky, Joseph (1975) *Material Requirements Planning*. McGraw-Hill Book Company ISBN 0-07-047708-6.

[2] MRP has been standardized, taught and promoted by the American Production and Inventory Control Society or APICS *(http://www.apics.org)*.

Historically, MRP and dependent demand superceded the practice of managing inventory by re-order point (re-ordering when stock fell below a quantity setpoint). Using dependent demand, MRP deduced the actual time-phased requirements, which in many situations gave significantly better results.

[3] The Knowledge Level concept comes from Martin Fowler (1997) *Analysis Patterns*. Addison-Wesley ISBN 0-201-89542.

[4] Known uses of the extended total supply chain dependent demand pattern can be found at:
http://www.i2.com
http://www.manugistics.com
http://www.pepper.com
http://www.berclain.com

[5] Kanbans are explained in Ohno, Taiichi (1988) *TOYOTA PRODUCTION SYSTEM: Beyond Large-Scale Production*. Productivity Press ISBN: 0-915299-14-3.

[6] Constraint Logic Programming is explained at:
http://www.cs.washington.edu/research/constraints/

[7] *Specifications*, a pattern by Eric Evans and Martin Fowler, presented at Plop ’97, in the Proceedings: Washington University Tech. Report #wucs-97-34.
http://st-www.cs.uiuc.edu/~hanmer/PLoP-97/Proceedings/proceedings.zip