

# The Language of Supply Chains

Corey Billington  
Strategic Planning and Modeling

**A**round the world, manufacturers are looking for ways to characterize and refine the order fulfillment processes they have developed for their products. To supplement technical and analytical approaches, this article proposes an innovative technique that builds on the analogy of natural language. Here we develop and illustrate a descriptive grammar, if you will, for designing the supply chain that makes the best sense for the product. It can be refreshing as well as useful to think of the optimal supply chain in terms of syntax, semantics, and style.

In our view, a supply chain grammar would consist of syntactical units for tasks, activities, and processes, along with rules for combining them, while the semantics of the supply chain—“the best sense for the product”—would be the value created by achieving maximum efficiency. Our fundamental point is that using excellent grammar in designing a supply chain ultimately has an *economic* meaning. The advantage of this approach is that we can analyze supply chains, examine their structure, develop new designs, share insights, select the design that creates the greatest value, and improve order fulfillment using a creative methodology.

In formulating a supply chain grammar, the most important syntactical feature is something we call the *push-pull boundary*<sup>1</sup>. This inflection point is where demand information—the actual customer order—exerts its influence on manufacturing. Inventory needs change dramatically with the position of the push-pull boundary, since some version of the product (or part) must be available to be processed and delivered when orders are received. Before this point, the product is *pushed* into a state of readiness in anticipation of customer orders. After orders are received, the product is *pulled* by demand—hopefully, with the speed and reliability needed to satisfy customer requirements. In terms of language, this is the point at which the audience for an utterance begins to determine its shape and meaning. The place where something pushed begins to be pulled by demand should be positioned to balance the costs of holding inventory against the need to serve customers quickly and reliably. Having a grammar enables us to revise the supply chain until the most economically advantageous position for the push-pull boundary has been found.

<sup>1</sup> A common term across Hewlett-Packard. Also called the “customer order penetration point” by theorists.

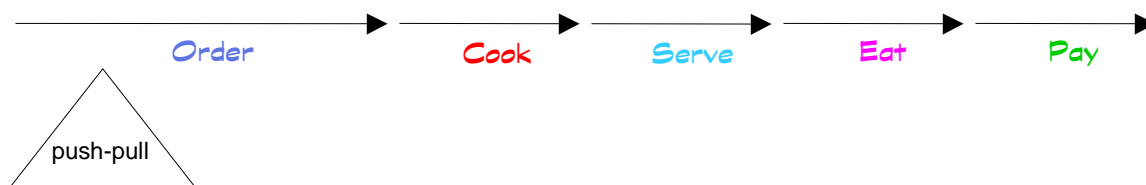
To illustrate the application of supply chain grammar, we develop a simple lexicon for restaurant types and use this language to describe different approaches to value creation in the industry. Then we present a more complex example from electronics—the manufacturing, localization, and delivery of printers by Hewlett-Packard. In each case, a standardized grammar enables a systematic exploration of order fulfillment characteristics. Moreover, the linguistic metaphor allows us to envision new structures and value delivery systems that may be more effective than those currently employed. Throughout this discussion, we draw upon suggestive conversations we have had with Brian T. Pentland<sup>2</sup>, Robert Sutton<sup>3</sup>, and others.

## The Lexicon of Fine Dining

Let's begin by considering fine restaurants, where customer preferences for quality and service are of paramount importance. An establishment wishing to compete on the quality of its cuisine would begin with a familiar series of processes, the lexicon of fine dining. Following Pentland, we can think of these lexical units as “moves” involving restaurant staff and customers. Of course, many moves are made in a sophisticated establishment, but at a basic level the lexicon can be broken down into the following verbs:

- **Order**—The customer makes selections from a menu
- **Cook**—The chef prepares and cooks the meal requested
- **Serve**—The restaurant staff serves the meal
- **Eat**—The customer consumes the meal
- **Pay**—The customer pays the bill.

On the following timeline, we see that most activities are performed *after* the order has been taken. Customization and service (measured by customer satisfaction) are hallmarks of the system, though customers must be prepared for a significant wait before the meal arrives. For better or worse, the push-pull boundary is positioned very early in the design of the supply chain:



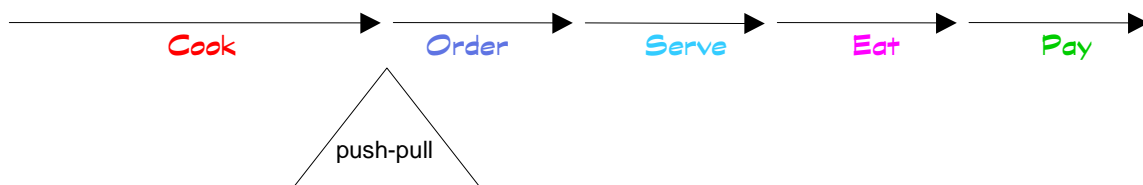
This grammatical sequence—Order, Cook, Serve, Eat, Pay—supports a system in which value is created through meal variety in response to individual requests; the classic *build-to-order* situation. However, there are opportunities for variation—for example, *process reversal* (a standard appetizer served before the customer orders) or *postponement* (pastries and toppings prepared in advance for custom desserts).

<sup>2</sup> Author of “Grammatical Models of Organizational Processes” (1995).

<sup>3</sup> Professor of Industrial Engineering, Stanford University.

Moreover, we may be able to change the push-pull boundary to enhance the order fulfillment and economic behavior of the supply chain. Alternative designs might create greater economic value and make the restaurant more competitive.

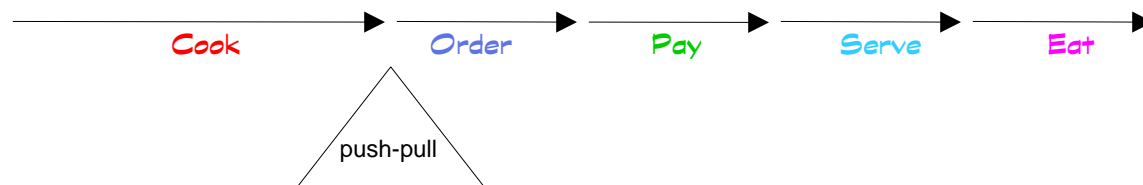
An interesting variation, for example, would be an establishment that prepares meals in anticipation of customers. Some restaurants specialize in cuisine—from prime rib to Peking duck—that takes more time to prepare than customers have time to wait. In such cases, the push-pull boundary for specific meals is different—since cooking is done before ordering—and inventory (in the form of cooked food) is put at risk. But a sufficient number of reliable customers for a specialized cuisine often justifies the sequence of Cook, Order, Serve, Eat, Pay. This reversal moves the push-pull boundary from the first position to the second, as shown below:



This new syntax often involves greater expense (more perishable inventory at higher unit cost) and more inventory held at risk (cooked food with a shorter shelf life than uncooked food). But the supply chain supports a value proposition rarely available, and there may be an advantage in cooking at a lower cost before customers begin ordering—and waiting. Many successful restaurants repeat variations on this theme to keep customers returning for the fine-dining experience.

## The Syntax of Fast Food

Against fine-dining establishments offering a high degree of customization and personal service (along with high prices), we can compare the fast-food industry. The most familiar category of quick-service restaurants can be characterized as “eat last”—a theme that applies to the entire menu. The syntax looks like this:

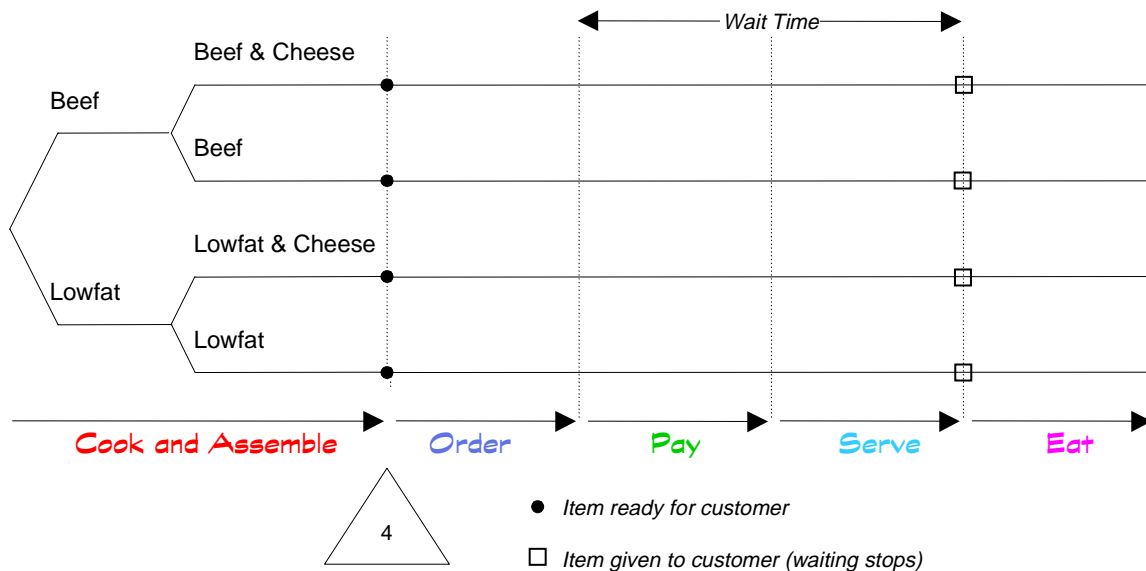


This approach involves not only cooking all meals in advance but disengaging from the customer as soon as possible. A clear example would be McDonald’s at noon. Later in the day the restaurant might return to the sequence of Order before Cook, but in all situations eating comes last. Note that the final transaction between the customer and the restaurant occurs *before* the meal is eaten. After payment, all that remains for the restaurant is to clean the tables and empty the trash—and by then the customer has disengaged from the supply chain completely.

The usefulness of supply chain grammar can be observed by extending this example. Many fast-food restaurants begin with the template just described but introduce variations to attract customers and keep them coming back. In all cases, however, the grammar and syntax of fast-food illustrate the importance of the push-pull boundary, especially its impact on efficiency, customer satisfaction, and expected profitability. To demonstrate the point, let's pursue the most efficient design for a restaurant that might satisfy the needs of a fast-moving business crowd.

## Burgers on Demand

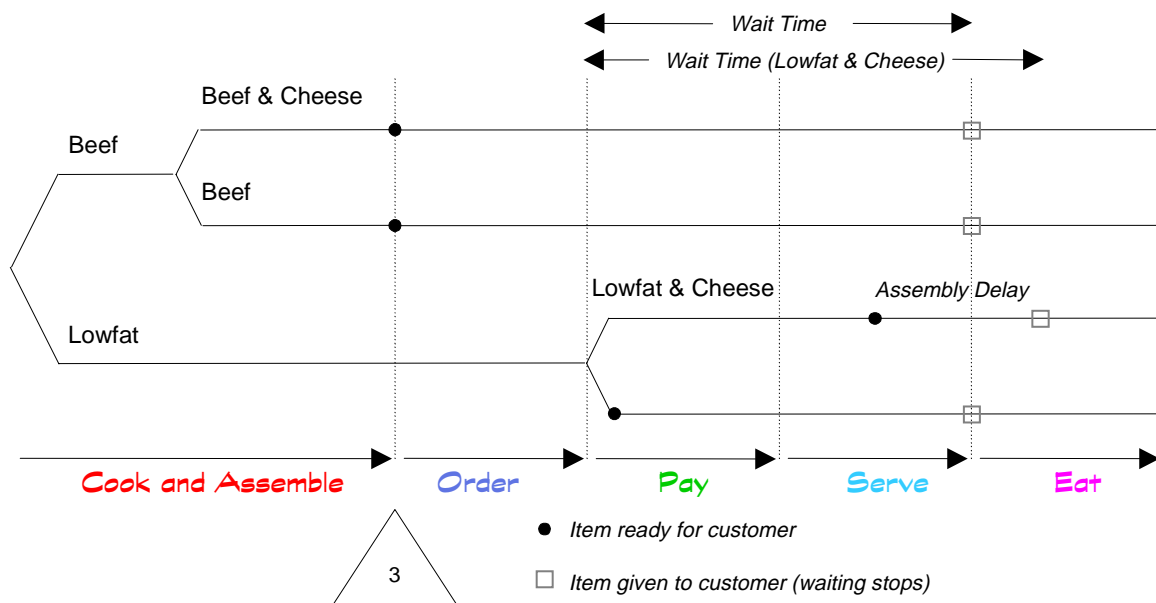
As an exercise in supply chain design, we begin working with the syntax of a place called Burgers on Demand. Our value proposition states that customers can enjoy any of four burgers on demand—hamburgers, cheeseburgers, low-fat (vegetarian) burgers, and low-fat burgers with cheese. As these meals are intended to be cooked and preassembled, the supply chain looks like this:



The exercise is like diagramming a sentence (and the weaknesses of the design appear as we examine its syntax). First, the push-pull boundary is placed where no less than four perishable items must be stocked and replenished continuously to satisfy customer orders. There are economic consequences to this approach—the more stock-keeping units needed before the push-pull boundary, the more inventory write-offs to be expected. Second, notice that all products must be assembled and customized before orders are received. Finally, the push-pull boundary initiates a process of ordering and paying that must be aggregated as *customer wait time*. Since people are very sensitive to the time they spend waiting, we can expect some customers to become impatient and avoid repeating the experience. As the diagram makes clear, this design has its limitations.

Looking for ways to improve the design, we notice that all manufacturing takes place before the customer orders. Patties for four burger types must be cooked in advance, and cheese must be added in two instances. Finally, sauces, condiments, and a bun must be added to each of four products before making them available to customers “on demand.”

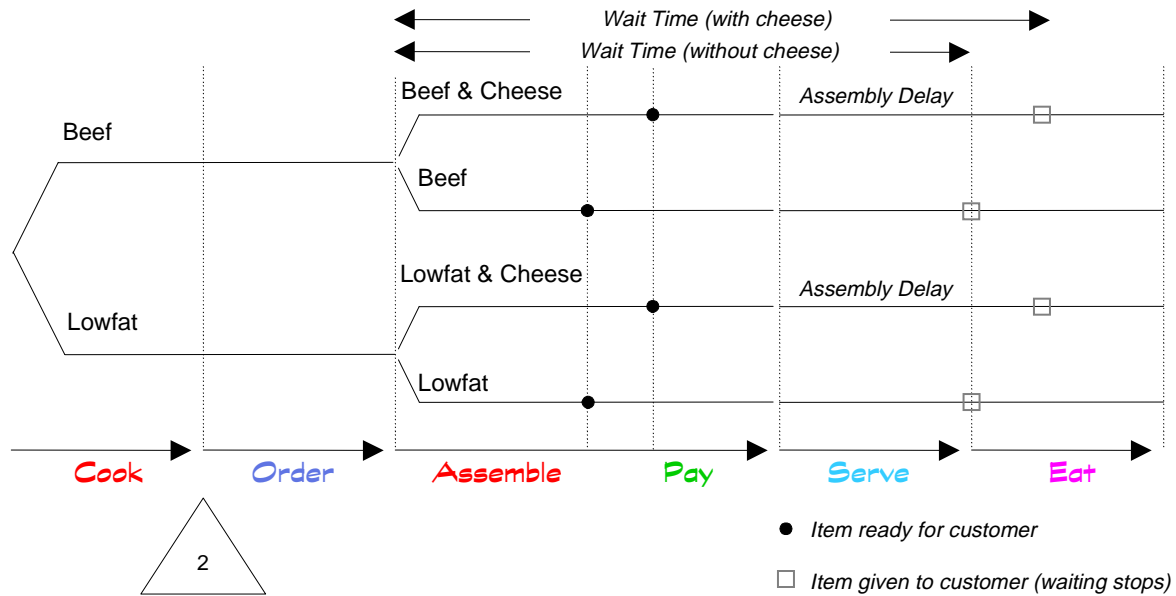
To reduce inventory requirements at the push-pull boundary, we keep ordering at the point where the patties have been cooked but delay the addition of cheese to the low-fat burger with cheese until *after* orders are received. Since cheese is high in fat, we realize that this may be our least popular selection. (Here we begin to appreciate how product design plays a crucial role in determining supply chain efficiency.) The advantage of these changes is that we stock only three items in inventory and postpone customization of one product until it has been ordered.



In this design, cheese is added to the low-fat burger after the push-pull boundary. Since the probability of orders for low-fat burgers with cheese may be low, we can tolerate a delay for orders including this selection. If the addition of cheese takes longer than the “pay” time, the customer wait time will increase for the fraction of orders that include the low-fat burger with cheese but remain the same for the majority of orders. (The potential delay in this process is indicated in the chart.)

Even minor modifications to the supply chain make a big difference, so we may need to describe its behavior mathematically. In this design, the average customer wait time for all orders equals the *Wait Time* multiplied by the probability that the order has no low-fat burger with cheese plus the *Wait Time (Low-Fat & Cheese)* multiplied by the probability that the order contains a low-fat burger with cheese. If the latter probability is low (as we expect), the average customer wait time for all orders will not increase significantly—and the inventory savings may make this option attractive.

Noticing the advantages, we consider further postponement opportunities—cooking the patties in anticipation of customer orders but delaying all other steps until after orders are received. We add Assemble to our lexicon to describe these tasks. Only two stock-keeping units are required in this design (cooked patties for the beef burger and low-fat burger); all other components are added after orders are received.



In this configuration, the average customer wait time increases because the number of orders that contain a low-fat burger with cheese or a hamburger with cheese is higher—note that wait time increases because the probability of receiving these orders has increased. It is important to compute the expected customer wait time through the entire supply chain to understand the costs as well as the benefits of holding fewer stock-keeping units and having fewer inventory write-offs. (Other consequences are associated with the cost of capacity, potential delays, and so forth.)

In an attempt to eliminate finished goods inventory, we might restructure the supply chain by moving “Order” before “Cook and Assemble,” making our burger joint resemble the fine-dining establishments discussed above. As might be expected, however, this approach involves the longest average wait time. Each customer must wait through the entire manufacturing and order fulfillment process, which includes cooking time for all burger types, before being served. As financial considerations drive us to cut inventories to the barest minimum, we see that increasing economy actually begins to jeopardize our value proposition. Along with the extended wait time, inventory requirements affect performance. Resequencing the “moves” in the supply chain—adding cheese to a low-fat burger after the push-pull boundary (*postponement*) and delaying paying until after the customer eats (*process reversal*)—has consequences our grammatical approach enables the supply chain designer to model and evaluate.

With this much experience, we may decide that Burgers on Demand is not where the greatest value can be achieved. Supply chain grammar would help us discover the all-you-can-eat cafeteria, for example, and study its economic behavior. With this approach, several cuisines are cooked in advance, the customer pays on entry, ordering and serving take place simultaneously, wait time is reduced to the time required to move through the line, and eating happens at the end. Here again, no transactions are needed after the meal has been assembled and paid for, and customers are free to exit the system at leisure.

## **Alternative Designs**

As we have seen, exploration of grammatical sequences helps us see the impact of the push-pull boundary. When ordering occurs before cooking, the system has little forecast error. And yet, any supply chain has difficulty anticipating what people will order, so designers must limit what can be ordered or lengthen the wait time for fulfillment. The grammar also helps us invent other restaurant configurations—for example, a sushi bar in which dishes are selected from little boats sailing past (and customers pay by their stack of plates). The supply chain designer need not experience such designs to consider the similarities and differences with other restaurant types.

With grammar, syntax, and meaning understood, new benchmarking approaches are possible for determining the competitive advantage of different supply chains. The lexicon can be enlarged to distinguish serving from transporting meals to the table or providing entertainment. Richer and more diverse restaurant types can be discovered by introducing symbols for major subassemblies, breaking off pieces for localization, and so on.

Leaving the restaurant industry, we should make our example more explicit. Cook is *Manufacturing*, Order is *Sales and Marketing*, Assemble is *Customization*, Serve is *Distribution*, Pay is *Accounts Receivable*, and Eat is *Customer Service and Support*. We can bridge from restaurants to electronics by considering their commonalities. As we proceed, we can enlarge our view using product variety trees and flow diagrams to represent product flow through the supply chain. These techniques allow us to study the consequences of the position we assign to the push-pull boundary.

## Supply Chains for Consumer Electronics

A descriptive grammar of supply chains can help us understand the dynamics of value creation in different industries. The complexities are enormous, of course, so we must simplify wherever possible. In consumer electronics, we can illustrate the approach by working through a portion of the supply chain for printers from Hewlett-Packard. What follows is a characteristic supply chain problem that shows how strategic alternatives can be evaluated to determine which does the most to increase value and decrease costs.

After the generic printer unit has been produced, the supply chain must include the most efficient assembly and localization processes for worldwide markets. Here are the major syntactical units to be included:

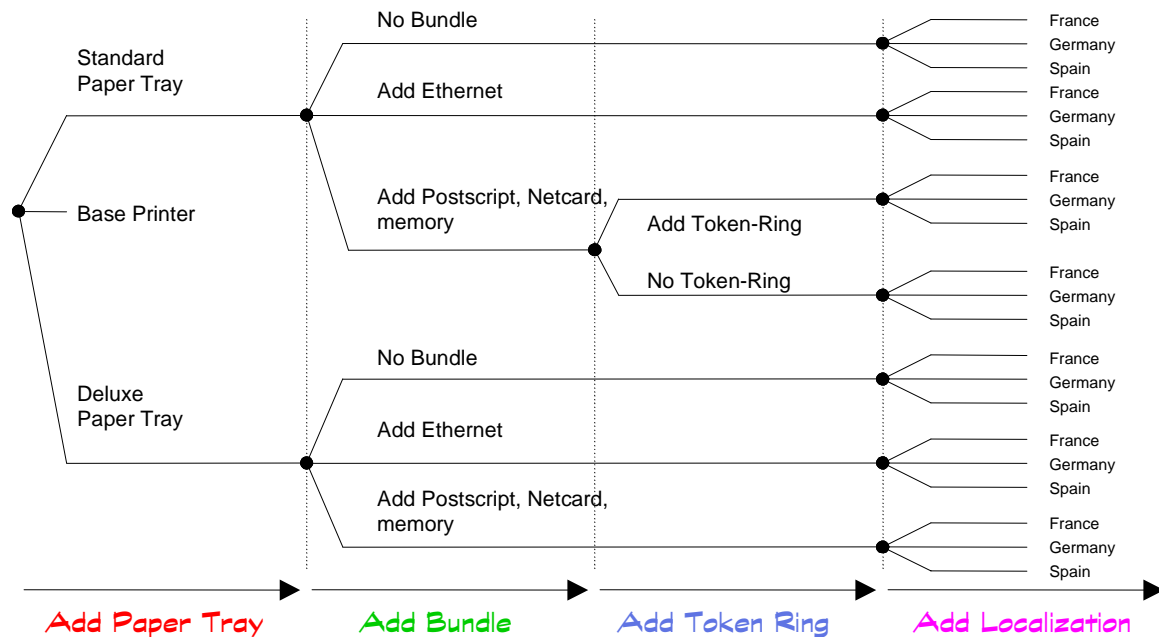
- **Add Paper Tray**—The paper tray may be standard or deluxe. The latter is an enhancement that enables standard paper tray to store more sheets of paper.
- **Add Bundle**—These hardware and software options are the Ethernet bundle, the Postscript bundle (with Netcard and memory), or none.
- **Add Token-Ring**—After testing, a Token-Ring unit can be added to the Postscript bundle. The options are Add Token-Ring and Test or Test only.
- **Add Localization**—At some point, there must be a process for language localization. For this example, our graphics will depict three options—French, German, and Spanish—but localization may involve hundreds of options.

Just as terms in any lexicon must be defined, we need to characterize and refine each of these processes to make it as efficient as possible. Economic considerations require us to decide where each process should be positioned. Should the process be performed *before* the push-pull boundary? Or *after*?

*Or do we need more analysis?* Sometimes it's not clear because characteristics of the process make it desirable on both sides. For example, a process might be both time-consuming and intricate (involving many options), so designers need to know whether the part involving many options can be separated from the part that takes a great deal of time. It might be possible to put some of the process on one side of the push-pull boundary and some on the other. Moreover, customers hate to wait for their order to be fulfilled, so we would like processes after the push-pull boundary to be as short as possible. Finally, factories prefer to manage the least number of items, so processes that create many variants should be scheduled after the push-pull boundary, if possible. As we think through the economic ramifications for the supply chain, decisions about which processes should be accomplished *before* and which *after* the push-pull boundary are crucial.

## Building a Template

Having defined lexical units, we can save time by building a template for the general structure of the supply chain. Quantitative analysis to determine the position of the push-pull boundary will depend on having alternatives derived from this template. The following diagram puts the “moves” in a reasonable order and depicts the material flows required:



Not included here are estimates of wait time and inventory requirements for the tasks to be accomplished. Because these activities are complex, we must work further on the lexicon. In other words, more refined definitions must be found for the processes included in the supply chain.

## Process Characteristics

Economic behavior constitutes the “meaning” of the design, so we must define each process carefully and assess its potential impact on value. Actual behavior must be thoroughly understood before we can position the push-pull boundary. The following characteristics influence the value contributed by each process step. We need to consider the impact of:

- Wait Time**—How much does the time required for this process (including quality and rework time) increase wait time for the customer? Intervals of waiting may be long or short; the longer they are, the greater the likelihood that customers will lose interest or turn to a competitor. In general, long process steps in manufacturing or assembly are better positioned *before* and short process steps *after* the push-pull boundary.

- **Options**—The number of options introduced by the process is also critical. Localization, for example, may involve dozens of configurations, and the product will branch into that number of directions when the process is done. If this happens *before* orders are received, forecast error is much more difficult to manage. By contrast, if Add Paper Tray occurs before ordering, only two options must be managed in response to demand. Obviously, adding a paper tray is not nearly as value destructive as adding localization before the push-pull boundary.
- **Component Value**—What is the value of the components consumed? How much “price protection” will be needed to safeguard our partners against devaluation? If parts are costly (like paper trays) or highly perishable (like memory chips), the process should come *after* receiving orders, to minimize obsolescence costs.
- **Process Intensity**—Processes can be heavy manufacturing, light manufacturing, or touch-only. Process and test engineering may be required for heavy or light manufacturing, while touch-only may require none. In general, high-intensity processes should be positioned *before* the push-pull boundary and low-intensity processes *after*. Before this point, the inventory buffer isolates upstream processes from demand uncertainty, so tasks can be made more efficient, manufacturing can be linearized, and total supply chain costs can be minimized. It may also be possible to centralize global manufacturing before the push-pull boundary, so goods can be transported across oceans (possibly less expensive than replicating capacity in different parts of the world).

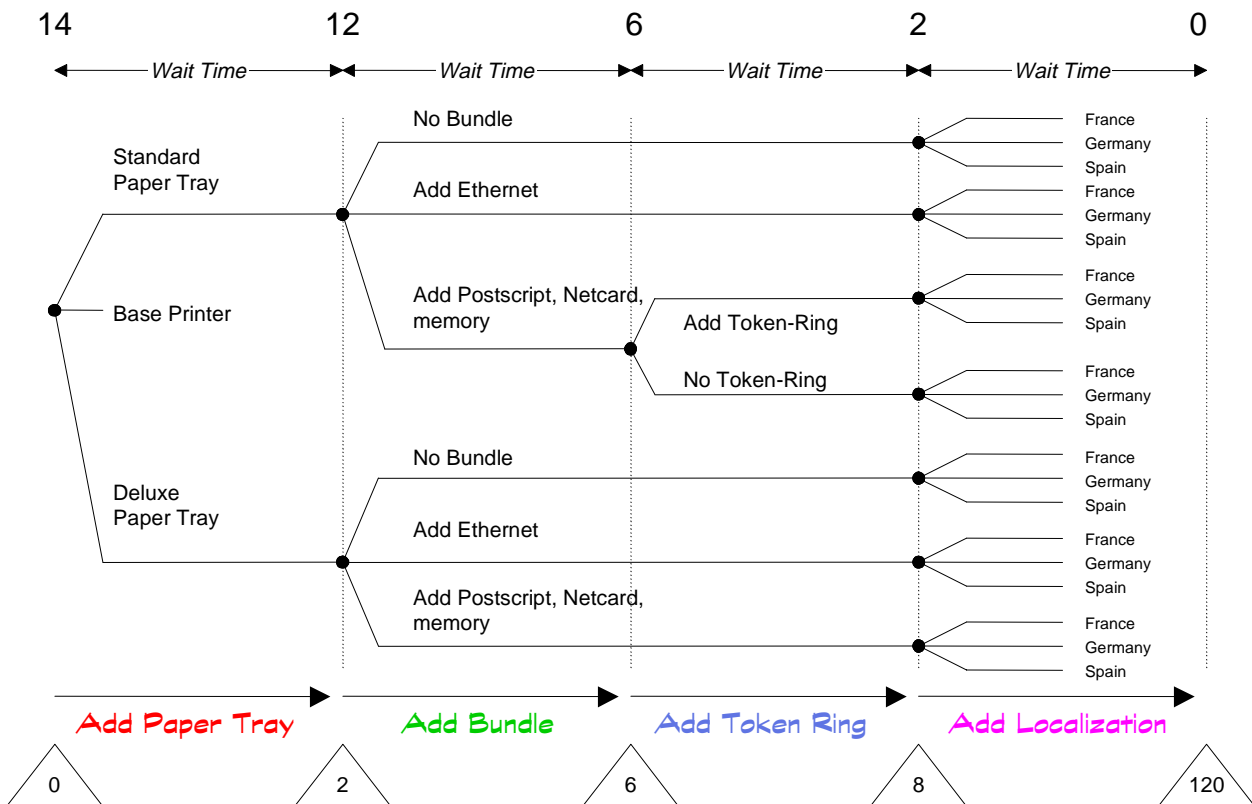
Because of their economic impact, these and other process characteristics must be reflected in the alternatives specified for analysis.

## What Are the Alternatives?

Although our template suggests many alternatives, the two most promising are depicted below. Triangles are shown with these grammatical sequences to suggest how inventories would change at each location of the push-pull boundary.

## Add Paper Tray First

The first alternative conforms to our template. Here we see the stock-keeping units needed at each potential location of the push-pull boundary (the numbers in the triangles). We also have estimates of the wait time customers must endure as a result of each process step. (To simplify, we ignore the time required for ordering, serving, and other activities).



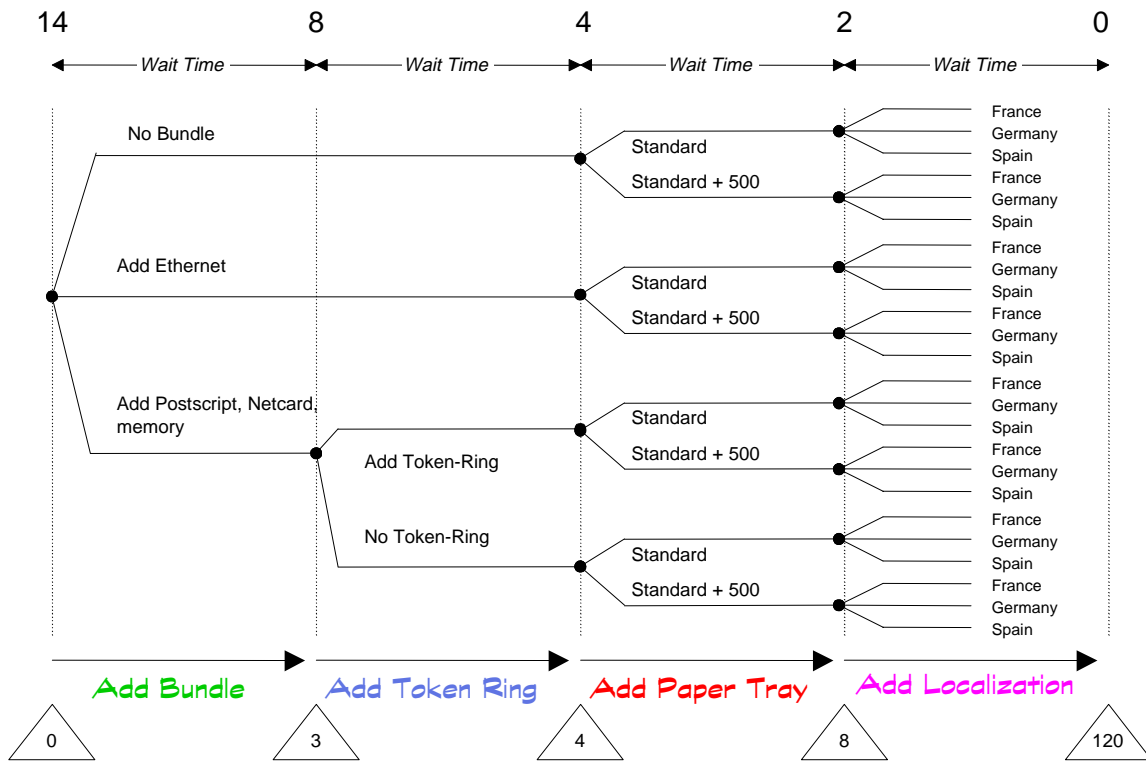
Scanning this diagram, we recognize trade-offs between wait time and inventory.

- With build-to-order (*left*), only generic parts are stocked so we minimize inventory (zero) and estimate substantial wait time (14). Again, these are our best estimates of the time and material required at this process step.
- With build-to-stock (*right*), fully localized products are built to forecast. Although there is no wait time, we need 120 different sku's in stock (and foresee substantial inventory requirements).

More interesting, however, are potential locations in the second, third, and fourth positions. When we model the economic behavior of this alternative, we need to understand the impact of holding 2 inventory items with 12 units of wait time (*second position*), 6 inventory items with 6 units of wait time (*third position*), or 8 inventory items with 2 units of wait time (*fourth position*). Along with other kinds of information (discussed below), these estimates must be available to the evaluation process.

## Add Bundle First

Process resequencing gives us our second alternative. In an attempt to improve the economics of the supply chain, we move Add Bundle and Add Token-Ring ahead of Add Paper Tray. Build-to-order (*left*) and build-to-stock (*right*) are the same as before, but the intermediate positions are different. The analysis needs to consider 3 inventory items with 8 units of wait time (*second position*), 4 inventory items with 4 units of wait time (*third position*), and 8 inventory items with 2 units of wait time (*fourth position*).



## Preliminary Evaluation

The objective of the evaluation is to find the supply chain design that minimizes the *total cost*. The following estimates help us perform a preliminary evaluation:

- Estimated Inventory-Driven Costs**—For every stock-keeping unit maintained before the push-pull boundary, we anticipate significant inventory-driven costs. We estimate these costs by multiplying the number of units by \$300,000.
- Estimated Lost Profits**—For every unit of wait time, we anticipate losing customer orders. This is a key driver. We might estimate, for example, that each unit of delay will cost \$500,000 in lost profits during the product life cycle.

Quick calculations provide insights that are consistent with our understanding of the processes involved. In both cases, the aggregate cost of inventory and wait time at Start (*first position*) or Add Localization (*fifth position*) make these unacceptable for the push-pull boundary. As the following tables show, neither build-to-order (\$7,000,000) nor build-to-forecast (\$36,000,000) would be cost-effective.

Push-Pull Boundary	SKUs	Cost	Wait Time	Cost	Aggregate
Start	0	\$ 000	14	\$ 7,000,000	\$ 7,000,000
Add Paper Tray	2	\$ 600,000	12	\$ 6,000,000	\$ 6,600,000
Add Bundle	6	\$ 1,800,000	6	\$ 3,000,000	\$ 4,800,000
Add Token Ring	8	\$ 2,400,000	2	\$ 1,000,000	\$ 3,400,000
Add Localization	120	\$ 36,000,000	0	\$ 000	\$ 36,000,000

Moreover, the lexical unit (or “move”) with the greatest change in significance is Add Paper Tray. This process contributes \$6,600,000 to costs in the first alternative, but its impact drops to \$3,400,000 in the second. The following table illustrates the effect of this process resequencing.

Push-Pull Boundary	SKUs	Cost	Wait Time	Cost	Aggregate
Start	0	\$ 000	14	\$ 7,000,000	\$ 7,000,000
Add Bundle	3	\$ 900,000	8	\$ 4,000,000	\$ 4,900,000
Add Token Ring	4	\$ 1,200,000	4	\$ 2,000,000	\$3,200,000
Add Paper Tray	8	\$ 2,400,000	2	\$ 1,000,000	\$ 3,400,000
Add Localization	120	\$ 36,000,000	0	\$ 000	\$ 36,000,000

While the trade-off between wait time and inventory is clear, a decision based on these results would be premature. Other trade-offs may change the cost structures shown above. To find the best alternative, the analyst must describe economic behavior more completely—taking into account such things as process intensity, component devaluation, price protection, and other factors to determine the cost implications of different approaches.

## Extended Evaluation

Extending our evaluation, we include the impact of linearization (process intensity) and price protection (component value) relative to the push-pull boundary. Here are descriptions of the trade-offs involved and estimates of their combined impact.

- Linearization**—Before the push-pull boundary, processes of different intensities—low, medium, or high—may be linearized for greater efficiency. Extra costs are incurred *after* the push-pull boundary because of capacity restraints and limited optimization possibilities. For this example, we estimate the following intensities: Add Paper Tray (low), Add Bundle (medium), Add Token-Ring (medium), Add Localization (low). To reflect added costs after the push-pull boundary, we estimate \$100,000 over the product life cycle for low-intensity processes, \$1,000,000 for medium-intensity processes, and \$10,000,000 for high-intensity processes.
- Price Protection**—If market conditions lead to falling prices for consumers, products waiting to be sold are protected and differences may be refunded to our channel partners. The cost of price protection depends on the value of components consumed by a process relative to the push-pull boundary. If an expensive component is added early, there is a greater chance that its value will be reduced before the product is sold. If this happens, price protection may kick in. If an expensive chip is used in the product, for example, ideally we want orders in hand before adding that chip (because its value decreases). To reflect price protection costs *before* the push-pull boundary, we estimate the following risk levels—Add Paper Tray (high), Add Bundle (medium), Add Token-Ring (medium), Add Localization (low). Our financial estimates are \$100,000 (low), \$500,000 (base case), and \$1,000,000 (high) for the value of components at risk before the push-pull boundary.

The following tables present the aggregate costs associated with linearization and price protection for the alternatives under consideration.

Push-Pull Boundary	Linearization	Cost	Price Protection	Cost	Aggregate
Start	medium	\$ 2,200,000	none	\$ 000	\$ 2,200,000
Add Paper Tray	low	\$ 2,100,000	high	\$ 1,000,000	\$ 3,100,000
Add Bundle	medium	\$ 1,100,000	medium	\$1,500,000	\$ 2,600,000
Add Token Ring	medium	\$ 100,000	medium	\$ 2,000,000	\$ 2,100,000
Add Localization	low	\$ 000	low	\$ 2,100,000	\$ 2,100,000

### Add Bundle First

Push-Pull Boundary	Linearization	Cost	Price Protection	Cost	Aggregate
Start	medium	\$ 2,200,000	none	\$ 000	\$ 2,200,000
Add Bundle	medium	\$ 1,200,000	medium	\$ 500,000	\$ 1,700,000
Add Token Ring	medium	\$ 200,000	medium	\$1,000,000	\$ 1,200,000
Add Paper Tray	low	\$ 100,000	high	\$ 2,000,000	\$ 2,100,000
Add Localization	low	\$ 000	low	\$ 2,100,000	\$ 2,100,000

### Complete Evaluation

At this point, we aggregate the “preliminary cost” (the trade-off between inventory and wait time) and the “extended cost” (other trade-offs involving linearization and price protection) to complete our evaluation. The following results allow us to compare aggregate costs at each potential position of the push-pull boundary.

Add Paper Tray First	Preliminary Cost	Extended Cost	Aggregate
Start	\$ 7,000,000	\$ 2,200,000	\$ 9,200,000
Add Paper Tray	\$ 6,600,000	\$ 3,100,000	\$ 9,700,000
Add Bundle	\$ 4,800,000	\$ 2,600,000	\$ 7,400,000
Add Token Ring	\$ 3,400,000	\$ 2,100,000	\$ 5,500,000
Add Localization	\$ 36,000,000	\$ 2,100,000	\$ 38,100,000
Add Bundle First	Preliminary Cost	Extended Cost	Aggregate
Start	\$ 7,000,000	\$ 2,200,000	\$ 9,200,000
Add Bundle	\$ 4,900,000	\$ 1,700,000	\$ 6,600,000
Add Token Ring	<b>\$ 3,200,000</b>	<b>\$ 1,200,000</b>	<b>\$ 4,400,000</b>
Add Paper Tray	\$ 3,400,000	\$ 2,100,000	\$ 5,500,000
Add Localization	\$ 36,000,000	\$ 2,100,000	\$ 38,100,000

The evaluation clearly indicates the most cost-effective location for the push-pull boundary—the third position of Add Bundle First (the design created with process resequencing). In our best sequence, Add Paper Tray and Add Localization are postponed until customer orders have been received. The grammatical sequence that makes “the best sense for the product” arises from an analysis of economic behavior made evident by the language of supply chains.

## Extending the Language

Many dimensions of economic behavior can be captured by extending the language of supply chains. The approach enables designers to create a unique lexicon for the individual product, quantify the implications of proposed “moves,” perform an economic analysis, and make decisions that reflect customer interests, product designs, manufacturing options, distribution networks, and service and support.

In suggesting “extensions” to the language, we want to emphasize the need for creativity on the part of designers. No less important than product design is the design of the supply chain by which it will be manufactured and delivered. Our diagrams have included graphics—icons, colors, arrows, and other features—to render the subtleties of design alternatives. With each new dimension, there is an opportunity to extend the technique by which economic performance is expressed. For the dimensions discussed below, we are developing the means to capture and communicate the nuances of effective supply chains. Here we see how excellence in supply chain design can be expressed through excellence in supply chain rhetoric.<sup>4</sup>

### Customer Dimension

We have already characterized customers by their willingness to wait for what they want. The language captures the wait time for any activity, aggregates wait time across the supply chain, and identifies trade-offs (especially with inventory required to respond to customer urgencies). The language can be extended to address such questions as:

- How do customer interests affect the performance of the supply chain?
- Should the supply chain be optimized to respond to customer concerns?
- What are the trade-offs between customer preferences and supply chain costs?
- Can the push-pull boundary be repositioned to the advantage of customers?

### Product Dimension

Product designs can be characterized by their suitability for different approaches to manufacturing, assembly, customization, and delivery. We have observed the economic advantages of postponement. Other aspects of product design can be reflected in the language, including a surprising range of functionality, modularity, aesthetics, connectivity, and other features.

- How do different product designs affect supply chain performance?
- Can design alternatives result in trade-offs across multiple dimensions?
- What are the implications of product design for the push-pull boundary?

<sup>4</sup> We mean “rhetoric” in the positive sense communicated by Robert Eccles in *Beyond the Hype* (1992), a seminal treatise on the ways rhetorical devices can be used to advance business enterprise.

## Manufacturing Dimension

Manufacturing processes can be characterized by their intensity as well as their suitability for linearization, automation, or other engineering approaches. The language allows us to register the impact of such processes and make the right trade-offs. Many considerations can be expressed, including value management of components and parts (with implications for price protection) and appropriate levels of testing, usability, and quality control.

- How do unique product designs influence supply chain performance?
- Can design alternatives result in trade-offs across multiple dimensions?
- What are the main drivers of manufacturing costs? How do they interact?
- How do manufacturing options affect the location of the push-pull boundary?

## Distribution Dimension

The language allows us to characterize a broad array of distribution networks, including order fulfillment processes at global, regional, or local levels using a variety of transportation options. Designers can make trade-offs across multiple dimensions, showing how customer preferences and product design alternatives, for example, interact with distribution options, including mass customization, channel assembly, and localization across the network.

- How do network capabilities interact with product design to affect performance?
- What trade-offs appear with different distribution network designs?
- How does the push-pull boundary change with different distribution options?

## Service & Support Dimension

Less obvious but highly important to the design of the supply chain are various approaches to service and support; these too can be reflected in the language. We can characterize alternatives in other dimensions by their impact on service and support, including warranties, trade-ins, and returns. Trade-offs with service and support costs should not be overlooked when working through the design.

- How do customer preferences or product designs affect service and support?
- What trade-offs are possible as a result of service and support alternatives?
- Does the push-pull boundary have service and support implications?

It will be evident that other dimensions—including advertising, marketing, and sales—can be rendered in the language we have been developing. In each case, the questions to be answered imply opportunities for trade-offs to support the most advantageous supply chain design for the product.

## Conclusion

We began by proposing that a language of supply chains would supplement other analytical approaches for designing and evaluating supply chains. We close by drawing attention to the obvious fact that supply chain design is an art (as well as a technical discipline) and deserves consideration for its extraordinary power to create and conserve value. The approach allows managers to explore how their supply chain contributes to the business, interacts with customers, and delivers the goods efficiently and effectively. Put into practice, the insights and understandings from these imaginative explorations can make the supply chain—and the company—far more competitive than before.

In particular, we think the position of the push-pull boundary—the point at which customer orders enter the system and change the forces at work—is one of the most important features the supply chain manager can control. The linguistic approach helps designers improve supply chain effectiveness through an imaginative exploration of alternative designs. By capturing and rendering influences across multiple dimensions and by revising the grammatical sequence through which these influences are expressed, designers can find the supply chain that makes the best economic sense for the product. Best of all, this innovative approach can be generalized across businesses and industries, so that the language of supply chains becomes a methodology widely shared.