

Applying supply chain optimization techniques to workforce planning problems

D. L. Gresh
D. P. Connors
J. P. Fasano
R. J. Wittrock

The IBM Research Division has developed the Resource Capacity Planning (RCP) Optimizer to support the Workforce Management Initiative (WMI) of IBM. RCP applies supply chain management techniques to the problem of planning the needs of IBM for skilled labor in order to satisfy service engagements, such as consulting, application development, or customer support. This paper describes two RCP models and presents two approaches to solving each of them. We also describe the motivation for using one approach over another. The models are built using the Watson Implosion Technology toolkit, which consists of a supply chain model, solvers for analysis and optimization, and an Application Programming Interface (API) for developing a solution. The models that we built solve two core resource planning problems, gap/glut analysis and resource action planning. The gap/glut analysis is similar to material requirements planning (MRP), in which shortages (gaps) and excesses (gluts) of resources are determined on the basis of expected demand. The goal of the resource action planning problem is to determine what resource actions to take in order to fill the gaps and reduce the gluts. The gap/glut analysis engine is currently deployed within the IBM service organization to report gaps and gluts in personnel.

Introduction

Supply chain optimization techniques have long been used to model the behavior of manufacturing supply chains in order to allow better planning and improve profitability and efficiency [1, 2]. With an accurate model of how raw materials are eventually turned into final products, one can gain insight into potential shortfalls of raw materials, inability to meet forecasted demand, and other problems. A model of the supply chain can include costs, substitution possibilities, time lags between sending an order to a supplier and the fulfillment of that order, demand expectations, production capacity limits, and other related factors.

Part of the mission of the Workforce Management Initiative (WMI) of IBM is to apply the same kind of analysis to the human resource supply chain. According to IBM, WMI is “a series of strategies, policies, processes and tools which enable optimal labor deployment built on a foundation of learning.” With better models of service

engagements and the human resources required to fulfill them, we hope that similar benefits can be realized. In many businesses today, including IBM, the human component of “production” is the most important and costly part of the supply chain, and it is critical to optimize the use of human capital. The Resource Capacity Planning (RCP) Optimizer was developed to allow a flexible framework to model the human resource supply chain.

While linear programming and other mathematical techniques have been used for some time to generate personnel schedules, as in, for example, call centers or airline shift assignments (see [3] for a review of optimization techniques for scheduling problems), the actual application of optimization techniques to the workforce planning environment is much less common, although the potential value is beginning to be recognized [4].

Part of modeling the human resource supply chain is creating “bills of material” (BOMs) for service

©Copyright 2007 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/07/\$5.00 © 2007 IBM

engagements, which are analogous to bills of material in the manufacturing process. A bill of material simply specifies how a final product, or a component in a final product, is to be built from a set of parts and capacities. For example, in the manufacturing scenario, a particular component may be built from a set of subcomponents (parts), and it requires some number of hours of machine time for assembly (capacity). Parts that are not used in a time period are available in the next time period, although the same is not true for capacity. Bills of material may include information about required quantities, time lags for acquisition or production, substitution possibilities or preferences, and other related considerations. In the human resource supply chain, bills of material typically specify the quantity of skilled personnel required in order to complete a service engagement, along with information about when in the engagement they are required. For example, for long engagements, different skills may be needed at different stages of the contract. Analogous to “bills of material” are “bills of product,” which specify what products are produced by a particular operation. In the manufacturing supply chain, this might be a set of components that are produced by a certain manufacturing step, which itself consumes some set of parts and capacities.

RCP is built using the Watson Implosion Technology (WIT) toolkit [5, 6], which consists of a resource capacity model, solvers, and an Application Programming Interface (API). This paper describes the kinds of problems that RCP addresses and discusses how some of the unique attributes of the human resource supply chain were incorporated into the RCP models. In the models described here, we use terminology from both the supply chain and the workforce-management domains. While some of the traditional supply chain terms such as *inventory* and *scrap* may seem out of place or even callous when applied to human beings, we found that the use of such terminology was often helpful in developing a useful analogy between supply chain management and workforce management.

We needed to solve two different kinds of problems, gap/glut and resource action. In the gap/glut problem, the goal is only to compute what the resulting “gaps” and “gluts” in resources would be, given the expected demand and the description of the resources necessary to fulfill the demand. This is similar to material requirements planning, or MRP, in the supply chain literature (see [7] for an explanation of MRP). However, gap/glut planning also differs from standard material requirements planning. MRP typically starts with information on the expected demand, and then uses the bills of material required for the demanded components in order to compute a list of required resources. Using information

on the supply on hand, MRP indicates “gaps” in the supply of necessary components.

Our customer for the RCP optimizer was the Labor Optimization organization within WMI. The mission of the organization is to develop business processes and decision support tools for managing the workforce of IBM. The users of the tools are resource capacity planners in the IBM Global Services business units who make decisions on how to deal with shortages and excesses of resources. Thus, our customer was interested in computing both the gaps and the gluts in human resources. The term *gluts* indicates areas expected to have a larger supply of people than will be used to fulfill demand. Note that gluts in standard industrial supply chains are not necessarily as important as in the human resource chain, since material excesses can be inventoried. However, the cost of “inventory” of human resources, that is, their salary, is usually significantly higher than the cost of holding excess physical inventory. Gap and glut information that relates to human resources can lead to improvements in business profitability by, for example, indicating areas in which retraining from one skill to another would lead to increases in efficiency.

Our model also differs from standard supply chain models in that the customer wanted to include a rather complex set of substitution possibilities in the model. That is, in many cases it is not necessary to “exactly” match a human resource to a job. Depending on the particular engagement, some amount of flexibility may be allowable. In addition, “what-if” scenarios may be explored, in which the impact of ignoring the effect of particular resource attributes on gaps and gluts can be investigated interactively. The decision-maker may discover, for example, that a strict requirement to match people to jobs within line-of-business boundaries has a large impact on overall efficiency. The wide variety of substitution possibilities made this problem different from the case of industrial production, in which typically only one part (or perhaps one from a small set of parts) is suitable for substitution. Our use of the WIT modeling library allowed us to incorporate both of these requirements—including substitution possibilities and exploring what-if scenarios—in our gap/glut analysis.

The second problem we address is the resource action problem. In this case, the model also incorporates descriptions of possible ways to resolve shortages and excesses, and can suggest the recommended course of action. The same substitution flexibility that is required for the gap/glut problem is also necessary for the resource action problem.

For each of these two classes of problems, we have two approaches for solving the problem. The first solution approach is priority-based: Given business rules on the use of resources such as preferences and priorities, a

heuristic algorithm implements the desired allocation rules. The second solution approach is based on linear programming, which is a method for optimizing a mathematically expressed objective given mathematically expressed equations and inequalities such as those ensuring that consumed capacity is less than available capacity. Given costs for using resources and taking resource actions, and rewards for meeting the demands, a mathematical programming model is formed and solved. Both of these solution methods are included as part of the WIT toolkit, which may be applied to the same WIT model once it has been built and has the necessary attributes defined. A precise statement of the problem addressed by WIT, including its formulation as a linear programming problem, is given in [5].

We make several assumptions in the models that are described here. First, the models are meant for intermediate-term planning purposes, of the order of three to six months in advance, and not for assigning specific people to particular jobs. Furthermore, our model does not consider workers on an individual basis, but rather groups of workers with matching skills and other attributes. Thus, people are modeled at an aggregate level, without consideration of such factors as planned vacation time. The granularity of description of supply and demand is at the weekly level. Second, people are described by job role, skill set, and other attributes; they are presumed not to change these attributes over the course of the planning horizon, although, as we later discuss, this is an area for future work. Third, we do not model personnel attrition.

The problem formulation of RCP treats demand as deterministic. The customer “filtered” the demand in the opportunity pipeline based on the probability of winning the service engagement deal, and only used information on highly likely engagements. This filtered subset of demand is considered to be deterministic in the RCP model. (The opportunity pipeline may be thought of as a view of potential opportunities for engagements with customers, including opportunities at all stages of development, from initial customer contact to signed contracts.) We have developed additional models that explicitly consider the demand uncertainties, but these models are not discussed in this paper.

Figure 1 shows the flow of data through the RCP engine. First, in order to populate the appropriate model, the engine reads data from flat files (files with records that contain variables separated by commas). The WIT “solve” method is then called on the model. Depending on the model, some post-processing of the data may be necessary, and the relevant output files are then created.

The WIT model for the basic RCP problem is shown in **Figure 2**. Note that the models shown in this document are subsets of the full model and highlight key points of

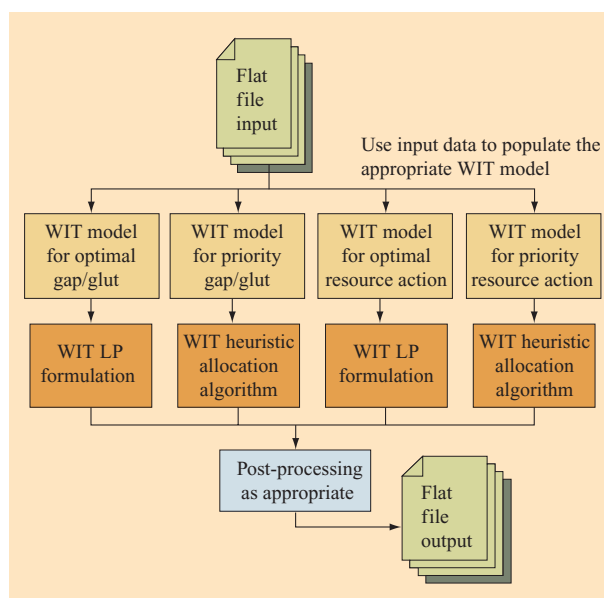


Figure 1

Flow of data through the RCP engine. The process builds a WIT model (yellow boxes) using provided input, and then calls a WIT “solve” method (orange boxes, which refer to either a linear program formulation or a heuristic algorithm) in order to determine the output, which addresses factors that relate to which demands are met or what substitutions are necessary.

interest for discussion. This basic model in **Figure 2** is augmented or simplified for each of the subproblems that we describe. We use the WIT notation to represent the elements of the model. A diamond represents a demand; a triangle represents a “capacity” part; an oval represents a “material” part; and a rectangle represents an operation, which converts parts into other parts. If a capacity part is not used within a time period, it becomes unavailable at the end of the period. On the other hand, a material part can remain in inventory for the next time period if it is not used in the current period. An example of a capacity part in the manufacturing scenario is available machine time. If a machine is not used one day, this does not mean that two days’ worth of machine time is available on the following day.

The portion of the model displayed in **Figure 2** shows a job or engagement, E_j (the triangle near the top of the figure). The demand for an engagement or job, denoted by *Demand: E_j* , is satisfied by the capacity part E_j . The capacity part E_j is produced by the operation *Build: E_j* . This is indicated by the presence of a bill-of-product (BOP) arc extending from operation *Build: E_j* to part E_j . The term “Build” simply means that the engagement will be produced (i.e., accomplished) by the collection of skilled people required as specified in the operation’s bill

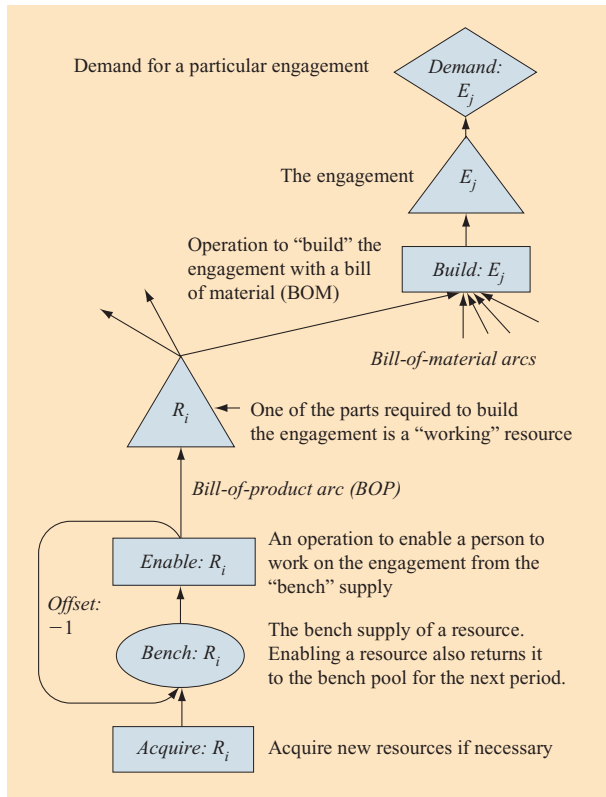


Figure 2

Basic WIT model for resource capacity planning. This diagram uses the symbols of the WIT modeling framework to describe the parts of the human resource supply chain. At the top of the diagram is the demand, indicated by a diamond, for a particular engagement. The demanded engagement is shown by a triangle. The rectangle below the engagement E_j indicates the operation necessary to build the engagement. This requires a number of different human resource types, only one of which (R_i) is shown. Another operation “enables” a person of this type to move from the “bench” in order to become a “working” person on the engagement. At the end of each period, the person returns to the bench in order to be available for another (or the same) engagement in the next period, which is indicated by “Offset: -1,” in which a value of -1 is assigned to the offset variable. If necessary, an operation exists that will “manufacture” new human resources, for example, by hiring.

of material. The operation *Build: E_j* may require several parts, including the capacity part denoted by R_i . For example, the operation may require five different resource types, e.g., a project manager, a tester, a programmer, and so on, only one of which is explicitly shown in detail in the slice of the model shown in Figure 2. The requested quantity of R_i (for example, five Java** programmers) is represented in the “consumption rate” (*consRate*) of the bill-of-material (BOM) arc from operation *Build: E_j* to part R_i . The consumption rate is not explicitly shown

in the diagram. Note that a BOP arc indicates an output of an operation, while a BOM arc indicates an input to an operation. The time period in which the resource is needed, for example three months after the start of the engagement, is represented by the *offset* of the BOM arc (also not explicitly shown in the diagram). Resources of the same type are often needed across many of the periods of the engagement. For example, the engagement requires five Java programmers in the first period, ten in the second, and ten in the third. Note that, as formulated, five programmers from the first period are not forced to be among the ten required in the second period, because we are concerned with a planning application rather than a personnel scheduling application and because people are modeled at an aggregate level. In the model, multiple BOM arcs exist from the engagement to the resource, with each BOM arc having the appropriate *consRate* and *offset*. (In Figure 2, for simplicity, only a single BOM arc is shown from the engagement to the resource.) The part R_i is produced by the *Enable: R_i* operation, which takes a part named *Bench: R_i* and produces the part R_i . We use the term “Bench” because we are using *available* people to accomplish a task, and these people are thought of as waiting “on the bench” and are not already doing other work. The *Enable: R_i* operation also produces the part *Bench: R_i* in the next time period. This is indicated by the “looping back” BOP arc extending from operation *Enable: R_i* to part *Bench: R_i* . The offset of -1 on the BOP arc indicates production in the next period.

The part named *Bench: R_i* is a material part; thus, unused inventory of this part is available for the next time period. In other words, if the resource is not “converted” to a working resource that is to be returned to the bench pool in the next period, it is automatically available for consideration for work in the following period. Finally, the part *Bench: R_i* can be produced, if necessary, by the operation *Acquire: R_i* . This models a hiring action or some other form of acquiring additional resources. Known hiring plans can be handled by the model. However, for the initial use of the model, data about future hiring actions was not available; thus, the model assumed that the supply of resources provided for the initial conditions remained constant throughout the planning horizon.

The human resource problem is fundamentally different from the standard industrial supply chain problems because the human resource problem requires a complex model in order to describe human resources and has a particular flexibility, described below, that is necessary to define substitution possibilities. For our application, the description of resources used both in the BOM of the engagement and in describing the available supply of resources is attribute-based and is driven by

parameters in a configuration file. Our customer uses the following attributes to describe his human resources:

- *Job role* (e.g., accounting system analyst, solution designer, database administrator. These are defined in an expertise taxonomy developed within IBM.).
- *Skill set* (e.g., technical editing, server consolidation, HR processes. Skill sets are also defined in the expertise taxonomy and are associated with particular job roles.).
- *Band: 1, 2, ..., 10*. The band level signifies the experience level of the employee, with higher-band employees more capable of independent work and leadership tasks.
- *Resource type* (e.g., IBM regular employee, subcontractor, joint venture).
- *Global resource flag* (identifies whether the employee is located in (for supply), or desired from (for demand), a country associated with low costs, such as China, India, or Mexico).
- *Region* (Asia Pacific; Latin America; North America; and the set of Europe, Middle-east, and Africa).
- *Country* (e.g., "U.S.," "Canada," "Japan").
- *State or province*.
- *City*.
- *Industry* (e.g., "industrial products," "media and entertainment," "automotive").
- *Line-of-business* (e.g., "e-business hosting services," "strategic outsourcing," "business transformation services").

Thus, an example of a fully qualified resource might be "database administrator, Oracle** database, Band 8, IBM regular, non-global, North America, U.S., Oregon, Portland, cross-sector, application management services."

Because the numbers and names of attributes are parameterized, it is simple to modify the model to handle different sets of attributes. To describe substitution possibilities, the user specifies which resource attributes must match between an engagement BOM and a human resource. For example, for a given run, the user can specify that it is not necessary to match the line-of-business attribute. The user can also define from-to substitution logic by giving substitutes for a specific attribute value. For example, the user can specify that it is acceptable to substitute a skill set of C++ programmer for a skill set of C programmer, perhaps with some cost or time lag for an optimization model or with some disfavor for a heuristic model. Substitutes are assumed to be consumed on a one-for-one basis with respect to the resource for which they are substituting, though this is not a requirement of the WIT model.

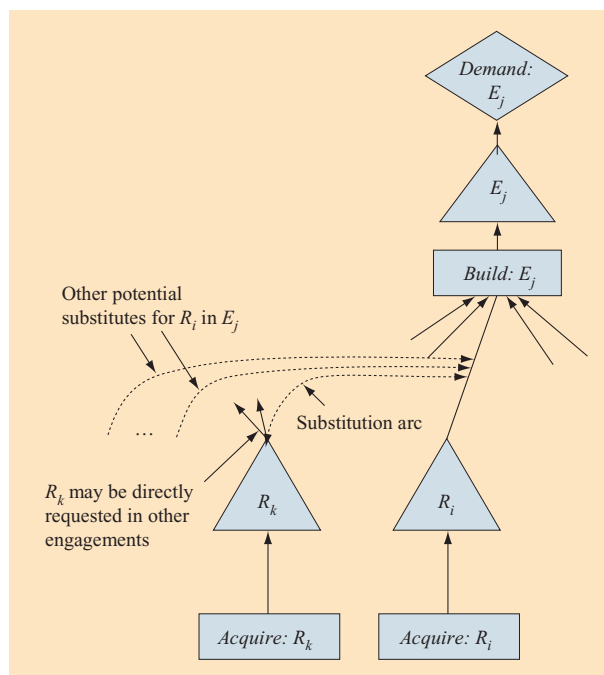


Figure 3

Priority-based gap/glut resource capacity planning model with substitution. In this simplification of the model shown in Figure 2, the *Acquire: R_i* operation is attached directly to the *R_i* part. In addition, we show the addition of a substitute arc indicating that a specific part *R_k* is allowable as a substitute for this particular BOM entry of *E_j*. This substitution may not be acceptable for other engagements, and by modeling the substitution at this point in the supply chain, the process remains flexible. Note that other allowable substitutes for *R_i* may exist in this particular BOM entry, and that *R_i* may be directly requested in other BOM entries or in other substitute BOM entries.

Internal preprocessing logic analyzes the user-specified matching and substitution rules and builds the appropriate substitution logic within the WIT model. Substitution arcs are attached to the engagement BOM arcs (Figure 3), allowing for the substitutions to be dependent on the engagement. For example, for one engagement it may be acceptable, even if not preferred, to substitute a subcontractor for a particular role, while for another engagement it is not acceptable. Example substitution arcs are shown in Figure 3 with dashed lines. If substitutions are independent of the engagement, we can attach the substitution arc in Figure 3 to the connector between the *Acquire* operation and the resource.

In our current formulation, people have only one skill. In the Conclusion, we describe extensions to the model that accommodate a more flexible view of a person's capabilities.

Gap/glut problem

The gap/glut problem may be solved with an MRP-like computation. We assume that all of the demand must be satisfied on time. Our goal is simply to compute the gaps and gluts of resources needed to fulfill the demand.

Optimal gap/glut

For the optimization-based gap/glut model, the basic model shown in Figure 2 is augmented with specific data as necessary to compute gaps and gluts that occur using the optimal solution. That is, if there are two types of human resource which can satisfy a particular demand, and there is more demand than supply, the model reports a gap on whichever resource is most advantageous to acquire. The bench supply of each resource R_i at period 0 is set equal to the external supply at period 0, i.e., the initial supply. Since this is an optimization-based model, we need to specify an objective function. WIT defines the objective function on the basis of cost and reward data attributes specified by the user. Thus, we set the *ShipReward* attribute for the demand *Demand*: E_j to be the revenue for engagement E_j . We set the *ExecCost* (execution cost) for operation *Enable*: R_i to be the monthly salary for resource R_i . We set the *StockCost* for *Bench*: R_i to be the monthly salary for resource R_i , because a person draws a salary even when he or she is not working on a project. Finally, we set the *ExecCost* of the *Acquire*: R_i operation equal to the realistic cost of hiring a new employee. We can also set the *offset* for the bill of product of this operation equal to the acquisition lead time, in weeks. The WIT modeling framework creates the appropriate linear programming formulation of the optimal gap/glut problem, assigning the costs, penalties, and objective on the basis of the data provided. This formulation is then run using a standard linear programming (LP) solver (e.g., COIN-CLP [8]) in order to determine an optimal solution, which in this instance is an appropriate allocation of human resources.

Priority-based gap/glut

In the priority-based approach for gap/glut, we simplify the basic model shown in Figure 2 by eliminating the *Enable*: R_i operation and the *Bench*: R_i part and directly connecting the *Acquire*: R_i operation to the working part (see Figure 3). The initial supply of human resources is now associated directly with the capacity part R_i . The enable operation and bench parts are in the basic model to address the subtleties of modeling human resources as WIT capacity parts and capturing the notion that a resource is either working or sitting on the bench. However, for the priority-based gap/glut problem, since our focus is on the net shortages and excesses of resources, we can simplify the model. The logic for allocating resources follows.

We assign priorities to the engagements and use the WIT heuristic allocation algorithm to determine the allocation of resources to engagements. This algorithm works with priority-style data rather than economic data such as costs and rewards. The priorities indicate the ordering in which the demands are to be met as well as the order in which alternative means of satisfying a demand (e.g., substitution) are to be considered. The allocation is built up sequentially on a “greedy” basis: Decisions made in earlier steps of the algorithm are not later reversed. The WIT heuristic generally computes its allocation quite rapidly compared with solving the corresponding problem as a linear programming problem; it has been used productively in numerous industrial supply chain applications, including problems involving more than 50,000 parts and 50 periods.

In the implementation that we produced for the workforce initiative of IBM, the engagements currently come in three varieties that we denote as Firm, Opportunity, and Forecast engagements, but this can easily be expanded. Firm engagements are those that are ongoing, or for which a contract has already been signed. Opportunity engagements are those for which a reasonably high probability of occurrence has been estimated, but which are not guaranteed. Finally, Forecast engagements are those that are simply expected to fill in the monetary gap between revenue from Firm and Opportunity engagements and the expected overall revenue for the planning horizon. The business process called for the resource planners to review the opportunity pipeline and select those engagements that the planner thought would become real engagements and then put them into the Opportunity pool. The business process suggested choosing those engagements that had odds of winning larger than some threshold. The engagements in the Opportunity pool were then considered by the model as deterministic but were treated with lower priority in the gap/glut calculations.

The business process recommended that the resource planners create a Forecast pool based on historical data for demands that had been realized but were not ongoing or in the opportunity system. Forecast demand is significantly different from Firm and Opportunity demand, since it is ultimately based simply on a dollar amount that must be generated in order to meet some target, and is not directly associated with specific engagements or potential jobs. Accurately predicting resource requirements due to Forecast demand is a separate problem which is not intended to be addressed by our modeling framework.

WIT provides powerful priority-based heuristics that allow demands to be prioritized by period. Our customer wanted all Firm demand, over the entire planning horizon, to be handled first, Opportunity second, and

employees are to be released. The offsets of the BOP arcs between the release operation and the release capacity part are set so that once a bench resource has been identified to be released, the release operation produces one resource for each period in the severance lead time in which the resource can stay in the system. The resource may work during the period, going through the enable-release operation, or “sit” and incur a “scrap cost” (a term from the supply chain literature) for the period. The monthly salary is incurred either by the execution cost of the enable-release operation or by the scrap cost of the released part.

The WIT framework allows us to set all of the relevant costs and rewards, including substitution penalties and time delays on substitution. For our application, we allowed the user to set penalties for each sort of substitution in the resource description, as well as to set a time lag for each attribute. For example, one could specify that substitution of “line of business 2” for “line of business 1” would incur a particular cost and time lag, while substitution of “city 2” for “city 1” would incur a different cost and time lag. Multiple substitutions are handled so that the overall time lag is the maximum of the individual attribute time lags, and the overall substitution cost is the sum of the individual attribute substitute costs. It is also possible to specify how much flexibility there is in the starting time of an engagement. For example, one may specify that a job is desired to be started in month 5, but that it can be started up to two months late, if necessary. (Late allowance is not part of the gap/glut analysis, because the gap/glut calculation is defined to be the excesses and shortages of resources, assuming that the demands are met on time.)

Once the description of the supply chain has been completed, we can compute an “implosion” of the problem in order to determine the optimal solution. Just as an MRP “explosion” starts with demand information and “explodes” the bills-of-material downward in order to determine the necessary supply, an “implosion” can be thought of as starting from the “bottom” of the supply chain model and determining the optimal allocations of parts to demands as well as the optimal set of actions to take, such as acquiring resources, releasing resources, or deciding not to fulfill some demands.

As an example, suppose that we have demand for several different engagements, which overlap in terms of the human resources needed. Suppose also that we do not have enough people to fulfill all of the demand for these engagements when requested. The optimal implosion algorithm produces different recommendations for action, depending on the particular monetary values assigned to different actions as well as the particular constraints of the problem. For example, for the case of very high revenue for

completing an engagement and relatively low costs for acquiring (hiring) new resources, the recommendation would be to hire. However, if the time lag for acquisition is very long or the costs very high, the algorithm may instead recommend that one of the engagements be delayed (if allowed) so that the human resource in short supply can work on one engagement, and when that engagement is completed, move to the second engagement. Alternatively, if the reward for one of the engagements is relatively low relative to the costs, or if delay is not possible, the algorithm may recommend that the lower-revenue engagement be declined. Some example input and output files for this scenario are shown in **Figure 5**. In this figure, for simplicity, resources are described only by job role and skill set. The data shown in Figure 5 represents a scenario in which there is insufficient on-hand supply to satisfy both clients at the time the engagements are requested. Because hiring costs are set relatively high in this example, and because one of the clients is flexible in the timing of the engagement, the optimal solution is to delay one of the engagements. Other models (for example, priority-based gap/glut) will have variations from these input and output file formats.

Clearly, in the case of many overlapping demands with different revenues, bills of material, substitution rules, etc., this kind of common-sense analysis becomes impossible to do manually.

Priority Resource Action

In the Priority Resource Action problem, we also call for an implosion to determine recommended actions, but in this case the possible actions are not given costs and rewards, but rather operate in a priority order. This is done using the WIT heuristic allocation capability. In contrast to the gap/glut calculations in which the shortages and excesses are computed on the basis of the planned supply of resources, in the Resource Action calculation we consider only the on-hand initial inventory of supply and compute the resource actions (e.g., acquire or release) that should be taken over the planning horizon. If we had to consider a planned supply line, RCP might have to make recommendations to release resources as soon as the planned supply was bringing them in. To eliminate these situations, we consider only the on-hand supply and make our resource action recommendations in light of the initial inventory.

The details of the priority-based resource action model are shown in **Figure 6**. As seen in the figure, resource R_i is modeled as a capacity part; thus, no inventory is carried over from any time period to the next. The supply of the capacity part R_i in every period is set uniformly to the initial inventory of resource R_i . Between operation *Acquire*: R_i and part R_i , one BOP arc exists for each

Engagements.csv

| Name | Revenue | Requested start | Late allowance |
|----------|---------|-----------------|----------------|
| Client A | 2000 | 0 | 0 |
| Client B | 3000 | 0 | -1 |

BillsOfMaterial.csv

| Name | Job role | Skill set | Period | Quantity |
|----------|---------------|-----------|--------|----------|
| Client A | App developer | Java | 0 | 2 |
| Client A | App developer | Java | 1 | 2 |
| Client A | App developer | Java | 2 | 2 |
| Client A | Architect | C++ | 0 | 3 |
| Client A | Architect | C++ | 1 | 3 |
| Client A | Architect | C++ | 2 | 3 |
| Client B | App developer | Java | 0 | 2 |
| Client B | App developer | Java | 1 | 2 |
| Client B | App developer | Java | 2 | 2 |
| Client B | Architect | C++ | 0 | 3 |
| Client B | Architect | C++ | 1 | 3 |
| Client B | Architect | C++ | 2 | 3 |

ActionCosts.csv

| Job role | Skill set | Acquire time | Acquire cost | Release time | Release cost |
|---------------|-----------|--------------|--------------|--------------|--------------|
| App developer | Java | 1 | 200 | 1 | 1 |
| Architect | C++ | 1 | 200 | 1 | 1 |

ResourceSupply.csv

| Job role | Skill set | Period | Supply | Cost |
|---------------|-----------|--------|--------|-------|
| App developer | Java | 0 | 2 | 2 |
| App developer | Java | 1 | 2 | 2 |
| App developer | Java | 2 | 2 | 2 |
| App developer | Java | 3 | 2 | 2 |
| App developer | Java | 4 | 2 | 2 |
| App developer | Java | 5 | 2 | 2 |
| Architect | C++ | 0 | 3 | 2 |
| Architect | C++ | 1 | 2 | 2 |
| Architect | C++ | 2 | 2 | 2 |
| | | | | |



Client B engagement
is delayed (as allowed)

Result

| Name | Fulfilled | Requested start | Actual start |
|----------|-----------|-----------------|--------------|
| Client A | 1 | 0 | 0 |
| Client B | 1 | 0 | 3 |

Figure 5

Input and output example files for the optimal resource planning formulation. The top four tables represent the inputs to the model, describing the potential engagements, their bills of material, the available supply of people, and costs associated with acquiring new resources. At the bottom is the optimal result based on these inputs, which is to delay the initiation of the engagement with Client B for three periods (as allowed by the late allowance value of -1 in Engagements.csv, indicating that any delivery date is acceptable). With other input data, recommendations might instead be to hire new workers or to decline an engagement. (CSV: comma-separated variable, a standard format for spreadsheet input.)

period. The offsets on the BOP arcs are set so that each unit of execution of operation *Acquire*: R_i in a period results in one unit of production (i.e., new availability) of part R_i in that period and in all subsequent periods. This model causes the heuristic of WIT to compute a solution that acquires the exact amount needed in each period. Once the heuristic implosion of WIT has been computed, we then post-process the output quantities in order to determine the acquisition and release recommendations. The acquisition amount per period is derived from the *execVol* variable (execution volume) of the *Acquire* operation, and the release amount per period is derived from the *residualVol* (residual volume) of the R_i part. We also mandate a policy not to release a resource if it is needed later on in the planning horizon.

Conclusion

In this paper we have presented the WIT models that we created to support the service business resource capacity planning process of IBM. The models address two planning problems, gap/glut and resource action. For each of these planning problems, we give two solution approaches, a priority-based and an optimization-based approach.

Depending on the availability of economic data or on the decision-maker's objectives or preferences, a priority-based approach may be either more or less appropriate than an optimization-based approach. Typically, optimization-based approaches require more complete economic data on all of the relevant costs, such as salaries, severance or hiring costs, and engagement revenues. Optimization models are often not accepted

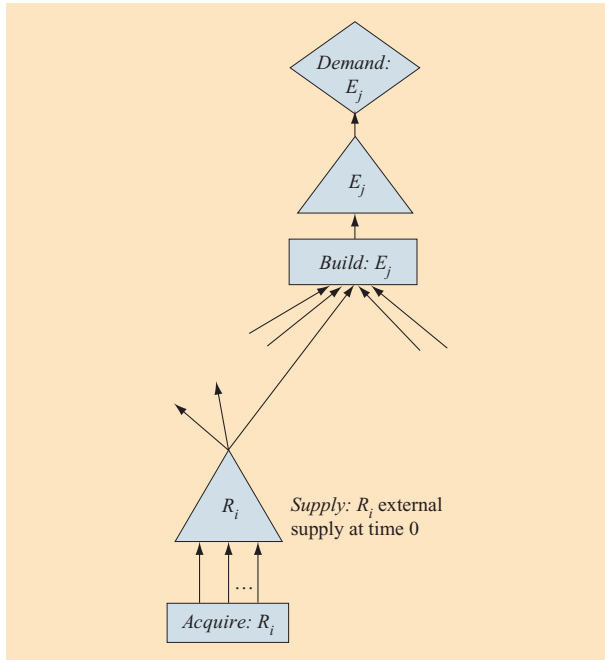


Figure 6

Priority-based resource action planning model.

by business users because it is difficult to assign values to nonquantifiable costs, such as the potential good will lost by using a less capable substitute instead of the best candidate for a job. Heuristic models, while not “optimal” in a mathematical sense, are easier to understand, and more closely follow the human method of assigning preferences to different actions. As it has turned out, the priority-based gap/glut approach is the one that is now being used in IBM business processes. Novel WIT models were created to handle human resources and the IBM business rules for prioritizing the use of resources.

The business processes associated with resource capacity planning are evolving, and the flexibility of the model and the WIT modeling environment is a good platform for allowing iterative development. We continue to work with the WMI staff as they evolve the business processes to best meet IBM workforce-planning needs.

The solid modeling foundation of WIT allows us to model and solve the important workforce-management problems involving gap/glut analysis and resource action planning. Both of these problems could be mapped quite naturally to the WIT modeling framework. The human resources themselves were not a direct mapping to either capacity or material parts, but WIT was sufficiently flexible that we were able to model them appropriately with no changes to the underlying WIT API.

We have also found the basic RCP problem framework to be extensible to other workforce applications, in which the basic concepts of engagements, attribute-driven resource descriptions, substitution allowances, bill of material descriptions, etc. can be put together with minor adjustments in order to solve variants of the human resource capacity planning problem. Further work must be done to determine what other workforce management issues can be readily modeled and solved using traditional resource planning environments, such as WIT.

Obviously, people are not parts. People can have multiple skills, and they can grow new skills or forget old skills. We have begun to design models that incorporate the learning and transition process in a way that, for example, would allow a person to “become” a new skill while still working in a capacity with a current skill. It may be difficult to maintain meaningful information on all of the past skills that a person retains with the notion of “skill attrition” through time. We have acquired information on the typical skill-set overlaps that occur in our employees’ profiles in order to allow substitution, at some probabilistic level, from one known “primary” skill set to an inferred “secondary” skill set.

While our work models people in the aggregate, separate work [9] directly addresses the problem of finding an optimal match of individuals to jobs. In staffing engagements, an even more sophisticated framework might take into account the fact that a certain collection of skills is actually what is necessary, rather than a certain set of skilled people. That is, we may need project management expertise, Java programming expertise, DB2* expertise, and testing expertise, but it is not necessarily the case that this requires four different people. This is a significantly different scenario from the standard manufacturing supply chain problem, in which simple parts come together to make a more complex component, and more sophisticated models would be necessary to address this scenario. As we conduct future research, we will have to discover whether WIT-like models and tools can effectively handle these more complex issues, and whether extensions can be made to WIT-like models to broaden the effectiveness of the modeling environments.

*Trademark, service mark, or registered trademark of International Business Machines Corporation.

**Trademark, service mark, or registered trademark of Sun Microsystems, Inc. or Oracle Corporation in the United States, other countries, or both.

References

1. A. G. de Kok and S. C. Graves, Editors, *Supply Chain Management: Design, Coordination and Operation*, Elsevier, Amsterdam, The Netherlands, 2003.
2. S. Voss and D. L. Woodruff, *Introduction to Computational Optimization Models for Production Planning in a Supply Chain*, Springer, Berlin, 2003.

3. A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff Scheduling and Rostering: A Review of Applications, Methods and Models," *Euro. J. Oper. Res.* **153**, No. 1, 3–27 (2004).
4. J. A. Ausink, R. Clemence, Jr., R. Howe, S. E. Murray, C. Horn, and J. D. Winkler, *An Optimization Approach to Workforce Planning for the Information Technology Field*, Rand Monograph Report MR-1484-A, 2002; see http://www.rand.org/pubs/monograph_reports/MR1484/.
5. R. Wittrock, "An Introduction to WIT: Watson Implosion Technology," *Research Report RC-24013*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 2006; see <http://domino.research.ibm.com/library/cyberdig.nsf/papers?SearchView&Query=RC24013>.
6. B. Dietrich, D. Connors, T. Ervolina, J. P. Fasano, R. Lougee-Heimer, and R. Wittrock, "Applications of Implosion in Manufacturing," *Supply Chain Management on Demand*, C. An and H. Fromm, Editors, Springer, New York, 2005.
7. T. E. Vollmann, W. L. Berry, and D. C. Whybark, *Manufacturing Planning and Control Systems*, 3rd Edition, Irwin/McGraw-Hill, New York, 1992.
8. "Computational Infrastructure for Operations Research (COIN-OR)"; see <http://www.coin-or.org/>.
9. Y. Naveh, Y. Richter, Y. Altshuler, D. L. Gresh, and D. P. Connors, "Workforce Optimization: Identification and Assignment of Professional Workers Using Constraint Programming," *IBM J. Res & Dev.* **51**, No. 3/4, 263–279 (2007, this issue).

Received September 15, 2006; accepted for publication October 15, 2006; Internet publication May 10, 2007

Donna L. Gresh *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (gresh@us.ibm.com)*. Dr. Gresh received her B.S. degree in engineering in 1983 from Swarthmore College and her M.S. and Ph.D. degrees in electrical engineering in 1985 and 1990 from Stanford University, where she studied the rings of Uranus using data from the spacecraft Voyager. She joined the IBM Thomas J. Watson Research Laboratory as a Research Staff Member in 1990 and spent twelve years conducting research in scientific and information visualization. Since 1992, Dr. Gresh has been a member of the Mathematical Sciences Department, with research interests in the area of workforce optimization.

Daniel P. Connors *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (dconnors@us.ibm.com)*. Dr. Connors received his B.S.E. degree in electrical engineering from the University of Michigan in 1982, and his M.S. and Ph.D. degrees in electrical engineering from the University of Illinois in 1984 and 1988, respectively. Since 1988, he has been a Research Staff Member at the IBM Thomas J. Watson Research Center. Dr. Connors has worked on modeling, simulating, and designing business processes and developing decision support tools for manufacturing and supply chain logistics. He is a member of the Mathematical Sciences Department at the Research Center, where he is currently working on developing business processes and workforce management optimization tools.

John P. Fasano *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (jpfasano@us.ibm.com)*. Mr. Fasano is a Senior Technical Staff Member and Manager of the Optimization and Mathematical Software group in the Mathematical Sciences Department at the IBM Thomas J. Watson Research Center. He received a B.S. degree in computer and system engineering in 1978, an M.S. degree in computer science in 1979, and an M.S. degree in operations research and statistics in 1983, all from Rensselaer Polytechnic Institute. He joined IBM in 1979. Before joining the IBM Research Division in 1992, Mr. Fasano worked in several IBM development organizations that had responsibility for the operating system of an early IBM Intel-based machine, vectorizing compilers for numerically intensive applications, and the development of Optimization Subroutine Library (OSL). In 1990, he taught for a year in the Mathematics Department at Grambling University. Mr. Fasano's current interests include supply chain optimization and open source math programming software tools. His current work includes reverse logistics and service personnel planning.

Robert J. Wittrock *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (wittrock@us.ibm.com)*. Dr. Wittrock is a Research Staff Member in the Department of Mathematical Sciences at the IBM Thomas J. Watson Research Center. He received his Ph.D. degree in operations research from Stanford University in 1983. He joined the IBM Research Division that same year and has generally worked in the area of applying optimization and heuristic techniques to supply chain management problems. He currently has primary responsibility for the ongoing design and implementation of the resource-constrained planning software known as Watson Implosion Technology.

Copyright of IBM Journal of Research & Development is the property of IBM Corporation/IBM Journals and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.