CRP – a Computing Engine for Resource Planning

David Y. Y. Yun, Professor of Electrical and Computer Engineering
Holmes 492, University of Hawaii, 2540 Dole Street, Honolulu, Hawaii 96822
Phone: (808) 956-7627; Fax: (808) 941-1399; E-mail: dyun@spectra.eng.hawaii.edu

Resource management is an intellectual process permeating daily human activities and important business decisions [Wilk88]. ***Planning*** for resource utilization and ***scheduling*** interrelated efforts are the necessary steps for solving these problems in order to achieve predefined goals under predetermined conditions. These resource management problems can, and often do, reach such complexity that require exceptional skills for judicious use and clear tracking of the resources. The key lies in the sharing of available resources to satisfy all the competing demands and restrictions on their use. Due to the diverse range of these problems, they are usually tackled one problem at a time, with little attention given to shared characteristics (among problems), generic techniques (across code modules), or common approaches (of analytic thinking) for this important class of problems.

A broadly applicable ***planning and scheduling*** methodology supported by a computer executable "***engine***" is presented here. The primary aim is to assist problem solvers with not only the general guidelines for characterizing their problems into systematic types and patterns but also transforming them into a general resource management paradigm where an embeddable, executable, work-horse module can assist to yield solutions. The ***Constrained Resource Planning (CRP)*** model, originating from work of Keng and Yun [Keng88, K&Y89, Yun92], has already been firmly established as a broadly applicable technique to solve numerous resource management problems (see Tables 1 and 2). These problems all deal with ***managing*** a combination of ***resources*** categorized as ***time***, ***space***, ***people***, and ***material***. Common to these problems are six essential concepts of the CRP model: ***resource, task, constraint, solution, criticality, and cruciality***.

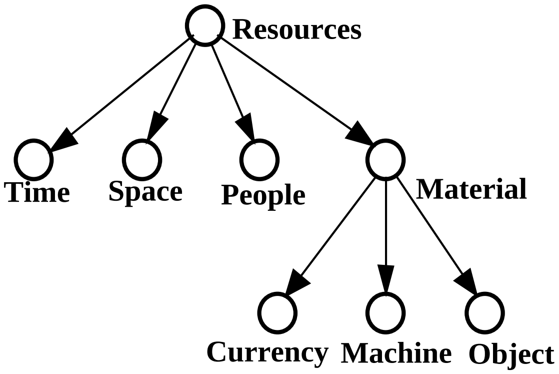

Figure 1. Resource Categorization

Resource is the collection of usable elemental units to perform the tasks of a given problem. Resources can be organized into four basic categories: ***time, space, people***, and ***material,*** which can be further separated into objects, machines, cost, etc. ***Tasks*** are work to be done in order to accomplish the overall goal of a resource management problem, using a specific amount of available, but often constrained, resource units. The tasks are usually interrelated due to competition among multiple tasks over the scarcity of resources. During the process of CRP problem solving, the active tasks are those still to be performed using the remaining resources. The set of active tasks is called the ***task agenda***. The identification of the most appropriate task to perform first is a critical decision.

A ***solution*** to a task is a collection of resource units assigned to perform that task. There may be alternative resources suitable for the performance of a specific task. All possible solutions of a task form the ***solution space*** for that task. The selection of available resource units among alternatives is the crucial resource management decision that determines the most appropriate solution. Relationships among different resources, solutions, and tasks are specified by ***constraints*** in resource management.

CRP model provides two domain-independent guiding principles for the identification of tasks and for the selection of solutions [HYH85]. The tasks are evaluated by a ***criticality*** function in order to apply the ***most-constrained strategy***. To the first order of approximation, the criticality function estimates the number of possible, valid solutions for each task. The most-constrained strategy operates by identifying the task with the least number of possible solutions, or the most critical task. This strategy is a natural generalization of the simple heuristic: always tackle the task with only one possible solution first, lest all needed resource units are consumed by performing other tasks and thereby reducing its own chance of completion. Thus, the most-constrained task is one that has the least flexibility for delay, hence has the highest priority - criticality.

The other guiding principle in the CRP model is used to direct the selection of a solution from the solution space. This domain-independent strategy selects the solution, which allows the most flexibility for the other tasks still to be have a solution. Called the ***least-impact strategy***, it selects the solution of the current task by minimizing the impact to other tasks, thus maximizing the feasibility of completing all remaining tasks. The impact of a solution on other tasks is measured by the ***cruciality*** function, which calculates the reduction to the number of valid solutions for other tasks. Selecting the solution with the least cruciality allows maximum flexibility for other solutions to satisfy other tasks.

Based on these six fundamental concepts, the operational mechanism of the CRP model is the ***Four-Corner Loop*** iterative process (of Figure 2), constituting an ***executable engine*** that transforms one ***state*** (corner box) to the next in accordance to the ***actions*** (specified by edge labels).

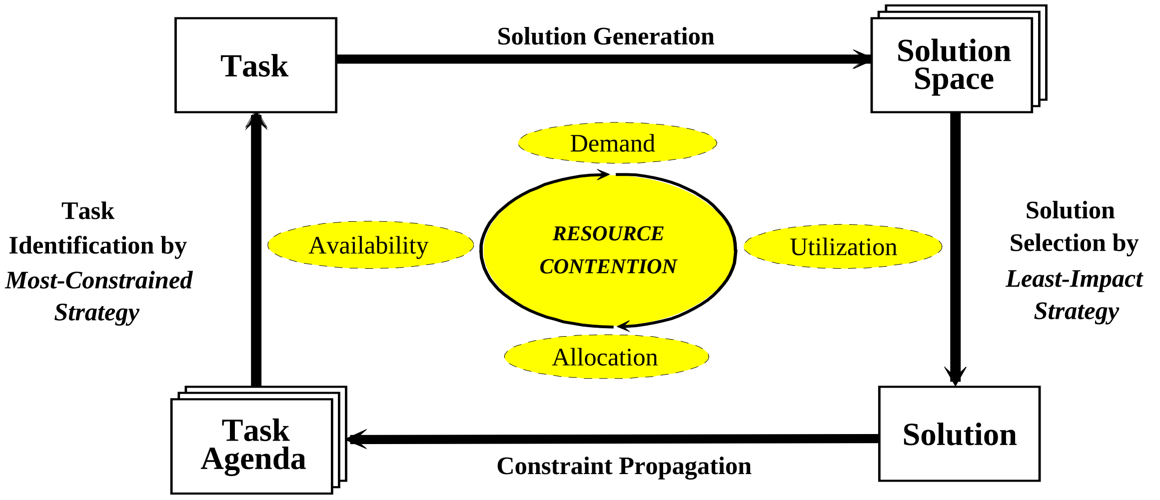

Figure 2. CRP Four-Corner-Loop

The CRP Four-Corner Loop has been repeatedly demonstrated to achieve superb efficacy by ***balancing*** the needs of ***using and sharing resources*** simultaneously under the requirement of completing all the tasks for a given problem. The task identification and the solution selection steps both compete over the available resources, thereby incorporating the natural compromising and economizing principles of resource sharing and utilization. Such ***global considerations*** at the ***local execution*** levels help to achieve the desirable delicate ***balance*** between "***task demand pull***" and "***resource supply push***". This balance allows the CRP engine to exhibit an uncanny ability to avoid backtracking (a common pitfall for most heuristic problem solvers) [KHY87b], so that at least one sub-problem is solved through each loop. When executing without backtracking, the CRP model achieves linear efficiency with respect to the number of tasks in the problem. Such economy, such efficiency, and such precision distinguishes CRP from many traditional AI (Artificial Intelligence) planning methods which could spend all the computational efforts “worrying” about the complex interactions among the sub-problems to an extent that nothing useful gets done. Thus, CRP proves its ability to serve as an intelligent planning and scheduling engine, creating suggested solutions to its users. Some CRP applications also keep score on the resulting plans and schedules, so as to provide a balanced user satisfaction index that allows further performance enhancement.

In the area of ***job-shop scheduling,*** CRP [KYR88] has already become an acknowledged ***performance benchmark*** [S&F90, Sadeh90]. The research community in business administration recently promotes theories and models of ***"resource based firm management"*** [Barn91a,b]. However, theories and models are not operational unless supported by an ***executable engine,*** as offered by CRP. The CRP model is a general decision support engine [KLY90, YLK89, Liu89], which possesses built-in intelligence through the criticality and cruciality heuristics, applicable to numerous domains. A prototype system incorporating the ***CRP Intelligent Engine***, was on exhibit at **COMDEX '92** and was a nominee of a **Byte Magazine** best new software award.

Though still a heuristic-based method, CRP has demonstrated consistent achievement of ***solution optimality*** in certain domains. One striking example arises from the well-known NP-complete Traveling Sales Problem (TSP) where solution optimality is difficult to determine [G&F92]. One sub-problem of TSP, called Diamond Lattice Connections (DLC), has provably optimal solutions that are simple to check [YQC92]. The CRP-based algorithm for solving the general TSP was directly applied to DLCs and produced optimal solutions for each size [YQC92]. This further demonstrates CRP's inherent robustness to attain optimal solutions, although it is well known that no heuristic algorithm can guarantee optimality.

Since resources utilized in different problems are primarily time, space, people, material, or combinations thereof, any new resource management problem can be mapped into a pattern of representative problems already solved with the CRP model. Table 1 identifies six representative application patterns, each involving two categories of resources. Each application pattern is a well-known pattern of planning and scheduling problem where typical resources characterize all applications within that pattern. Of course, since material has many distinct sub-types, it is possible to form an application pattern from two material types alone, such as cost and machine, without involving the other three types. Tables 1 and 2, though, focus on combinations of the top level categories of resources to give an already rich collection of application examples, illustrating the benefit of characterizing applications by their resource types.

Pattern No.	***Resource Type =*** **Application Pattern:**	***Time***	***Space***	***People***	***Material***
#1	**Production Optimization**	**X**			**X**
#2	**Space Utilization**		**X**		**X**
#3	**Occupancy Planning**		**X**	**X**	
#4	**Inventory Distribution**			**X**	**X**
#5	**Facility Reservation**	**X**	**X**		
#6	**Work-Shift Scheduling**	**X**		**X**	

Table 1. CRP Application Patterns

CRP – a Computing Engine for Resource Planning

Table 2 shows numerous solved resource management problems (see relevant papers) and classifies them according to their similarities to the application patterns of Table 1.

Application Pattern No.	#1	#2	#3	#4	#5	#6
Problem	**TM**	**SM**	**SP**	**PM**	**TS**	**TP**
1. Job-shop Scheduling	X					
2. Machine Utilization	X					
3. Income and Expense Management	X					
4. Process (Chemical, Production, ...) Planning	X					
5. Multi-processor Scheduling	X					
6. Floorplan Design (2D Packing)		X				
7. Pentomino (game)		X				
8. Steel Mill Production Planning		X				
9. VLSI Chip Placement and Wire Routing		X				
10. Shipping Container (3D) Packing		X				
11. Line-drawing Labeling		X				
12. 3D Image Feature/Object Recognition		X				
13. Map Four-Coloring		X				
14. Store Shelf Loading and Optimization		X				
15. Sub-Graph Isomorphism		X				
16. N-Queens			X			
17. Room (Hotel, Meeting, Class) Allocation			X			
18. Inventory Control				X		
19. Goods (Food, Tools, etc.) Distribution				X		
20. Weapon-Target Assignment				X		
21. Traveling Salesman Problem					X	
22. Itinerary Planning					X	
23. Telescope or Satellite Antenna Steering					X	
24. Diamond Lattice Connections					X	
25. Transportation (Bus) Dispatching					X	
26. Power Generator Maintenance Scheduling					X	
27. Nurse Scheduling						X
28. Service Shift Scheduling						X
29. Course (Teaching) Assignment						X
30. Work-load Assignment	X					X
31. Vehicle Path (Trajectory) Planning	X	X				
32. Supply Allotment and Scheduling	X	X				
33. Sheet Metal Cutting		X			X	
34. Robot Motion Planning		X			X	
35. Airport Gate Assignment			X		X	
36. Railway Train Scheduling	X				X	
37. Network Management	X				X	
38. Ocean Resource Management	X			X	X	
39. Conference Scheduling			X		X	X
40. Factory Automation Design and Control	X	X		X		X
41. War Games	X	X	X	X	X	X
42. Mission Planning	X	X	X	X	X	X

Table 2. Resource Management Problems and Classification

These solved problems, all using the CRP engine as the embedded work horse, demonstrates the inherent breadth of the CRP approach and the generic strength of this planning and scheduling tool for resource management. Comparable techniques, some of which already have marketable, computer-executable tools while others require more human intervention, include spreadsheets (SS), databases (DB), expert systems (ES), and operations research (OR) optimization techniques. Comparing several differentiating factors among these techniques with CRP, Table 3 offers some insight as to the practical values and general needs for a problem solving methodology together with an executable engine for resource management.

Differentiating Factors	**CRP**	**SS**	**DB**	**ES**	**OR**
1. Generic vs. Domain Specific	Generic	DomSp	DomSp	DomSp	DomSp
2. Solution Intelligence	Yes	No	No	Yes	No
3. What-if Scenarios	Yes	Yes	No	No	No
4. Planning/Scheduling Tool	Yes/Yes	Yes/No	Could	Could	Could
5. Ability to Optimize	Yes	No	No	Could	Yes
6. End-User must be Expert	No	No	Yes	No	Yes
7. User Mod of Constraints	Yes	Yes	No	No	Yes
8. Batch vs. Dynamic	Dynamic	Batch	Batch	Dynamic	Batch
9. Demand/Supply Balancing	Yes	No	No	Could	Could
10. Backtracking Reduced	Yes	n.a.	n.a.	No	n.a.

Table 3: A Differentiating Comparison with CRP

The methodology underlying CRP has also provided useful guidance for planners/schedulers faced with complex resource management problems [T&Y91b, AYT91, FTY92]. Usually, these problems involve limited resources, stringent requirements, conflicting constraints, and overwhelming data that confuse the problem solver. The experience of working through many different problems (in Table 2), dealing with all six application patterns (of Table 1), has revealed that a CRP-based way of thinking (approach) to sort out the complexities can be very helpful. CRP's resource categorization (Fig. 1), the six essential concepts (resource, task, constraint, solution, criticality, and cruciality), and the Four-Corner Loop process (Fig. 2) provide a useful conceptual structure to think through a complex problem in an organized manner. Often, intricacies in the problem can be clarified, confusing details can be organized, resource utilization and interrelationship can be simplified, conflict can be more easily resolved, and even useful heuristics can be generated and tested. Indeed, this captures the spirit of resource-focused management and problem solving.

The CRP model succeeds over traditional planning and scheduling techniques due to its balance between resource sharing and utilization, its separation of tasks and solutions, its sensitivity to interactions among them [KHY87b, KYR88], as well as its domain-independent control strategies that combine to effect both local execution objectives and global resource considerations. The result is a generally methodology both for "thinkers" by conceptually guiding the problem solving and for "doers" armed with an executable engine, together with the demonstrably improved effectiveness [Y&C93]. The model is embodied as a planning/scheduling engine and the methodology is supported by a set of tools that, collectively, offer a development environment available for industrial applications [Yun92].