# Chapter 24

Today is the last session I'm giving this academic year, and I'm going to break a tradition. Usually when I reach the end of the year I run out of steam, or even worse, I run out of interesting material. But not this time. This session we are going to cover the solution to the problem that has been haunting me for weeks. It's going to be a very special session. I even invited Johnny.

"Mark, will you please describe your environment to the class?"

Mark stands up and starts to fill the room with his deep voice. "As you know, we are no longer dealing with the A226. That modem is past history for us."

"It's current reality for our company." Ruth doesn't want anybody to misinterpret. "It's the biggest success we ever had."

"True," Mark smiles proudly at her. "Anyway, the three of us are charged with the task of shrinking the development time of . . . all development at Genemodem."

"Yahoo," Ted whispers.

"Right from the start we knew that the biggest challenge





would be to deal with resource contention." Mark pauses to find the proper way to explain the essence of their challenge.

"Haven't we already dealt with that?" Roger asks. "I thought that resource contention was solved by the concept of critical chain. What am I missing?"

If somebody has gone through a paradigm shift, it's Roger. He's dropped his facade of 'I know it all,' and developed a keen interest in the subject. From time to time he even seeks my help on implementation issues. That shouldn't be mistaken for a change in personality. He is still as cynical and self-centered as he was.

"Critical chain," Mark explains, "removes resource contention within a project. It does not address resource contention between projects at all."

"Why don't you use the same logic for a bunch of projects? What's the difference?" Ted doesn't see the conceptual difference between one project and many.

Before Mark is able to answer, I interrupt. "Ted, in your company you do work on more than one project at a time."

"Of course."

"So you must have enough intuition to answer your own question. Try. Give an educated guess. What might be the problems?"

"Offhand, I see a synchronization problem."

"Synchronization is an impressive word," I say. "So impressive that it's often used to disguise ignorance. Ted, you don't want us to suspect you of such a thing."

"No way." And he jumps to the other extreme. "Resource contention means that the same resource is supposed to do two different steps at the same time," he wastes our time defining a term that is clear to us. "Removing resource contention between two steps," he continues methodically, "necessitates, many times, postponing one of those steps. The problem is that, as we discussed at length, there is no clear way to decide which step to postpone. It is almost an arbitrary decision."

I like the way he's approaching it. In order to force him to

continue, I prod, "The same is true within one project. Why is it a bigger problem when the steps belong to two different projects?"

"Because two project leaders are involved," he confidently answers. "It's not like you work in one domain, where it doesn't matter which step you move. Here each project leader will naturally fight that the step to be postponed will not be his."

"Is it a big problem?" I continue my almost rhetorical questioning.

"Are you kidding?" Ted smiles. "Mark, now I see what you are facing. It's not just a synchronization problem, it's a nightmare."

"Fair description," Mark agrees. "Unfortunately we didn't realize how big the nightmare was before we stepped into it."

"With both feet," Ruth adds.

"Not because we are thick," Fred hurries to put in a caveat. "But because we didn't know what else to do."

"Do you want to hear what happened?" Mark asks.

I'm not the only one asking rhetorical questions.

"Well, the first problem was mechanical. Our projects, like almost any sizable projects, are described by about one hundred steps. You know, it takes some time to play with one hundred pieces of paper until all resource contentions are resolved. You move one piece to remove contention with one resource, and you have to move the other steps on the same path. This, many times, creates contention for other resources. It takes hours. Now, imagine doing it with six projects."

"So," Fred continues, "we went to our computer department."

"And that's the end of the story," Brian interrupts. "In my company everything you ask from the computer department takes months.

"It's the same in our company," Fred replies. "But we pulled rank. You see, our task is regarded as super, ultra, top priority. So we got a 'good-enough' piece of software from them pretty quickly. We loaded all the data, and then we started to play."



"And play. And play." Ruth laughs.

"These computers are an excellent way to procrastinate," Mark agrees. "We were dealing with minute contentions, things that if we had to do them manually, we would never have bothered. But we removed all contentions. Then, of course, as Ted predicted, we had to fight with all the project leaders."

"To cut a long story short, we squeezed agreement," Fred summarizes days of fierce arguments into one sentence.

"Then reality showed us what fools we were. Any speculations on what happened?" Mark asks the class.

Everybody is thinking hard, but nobody comes up with any ideas. Not even Ted.

Mark doesn't wait long. "Did you ever see a step that finished somewhat late?" He gives them a clue. "One small deviation in one step and BOOM—you get the domino effect, contentions all over the place. We found ourselves wasting all our time sorting out fights. Ted, you called it a nightmare? You are absolutely right."

"I can see it clearly," Ted agrees. "It could easily happen in my place. So what did you do?"

"Before we see the solution," I say, "can you tell us the conceptual mistake you made?"

"We were treating estimates as if they were real," Ruth answers.

"What do you mean?" I prod.

"Suppose a step was supposed to take ten days. You know there is a good chance it might take seven or fifteen, but we fed ten days into the computer. Then we treated this number as holy."

"I still don't understand," I say.

"We regarded contentions of three days as significant even when the estimate for the path was thirty days."

"Basically," Fred clarifies further, "we fell into the trap of thinking that eight times eight equals exactly sixty-four; we were trying to be more precise than the noise. Everybody was

fighting about contentions which, left alone, could have been easily absorbed by the buffers."

"As a result," Mark summarizes, "we were constantly changing schedules for nothing, and by that creating real problems."

"I understand," Brian says. "And thanks for telling us about it. Now I know what not to do myself, but I don't know what to do. We cannot ignore contentions.

"Absolutely," Mark agrees. "We cannot ignore contentions in one project, we definitely can't ignore them when we look at all projects."

"You see the dilemma we were caught in? Fred asks. On the one hand, we had to consider contentions, but on the other hand, when we did, we ran into the nightmare."

"So what did you do?" Brian is eager to find the answer.

"We called Professor Silver in to help."

"Which was totally unnecessary," I emphasize. "You knew the answer. You were just too lazy to realize it."

"That's unfair!" Ruth rebels. "Even after you showed us, it took me some time to digest."

"The answer was taught to you by Professor Fisher in your production course, and then it was elaborated on by Professor Wilson in your systems course."

I know that I'm unfair. I spent weeks working on it myself before I figured it out. But I want to open my students' minds to the possibility of transferring a good concept from one field to another.

"We are dealing here with resource contention problems," I start to explain. "Have you seen the same problem in your production course?"

"Of course," Brian says. "Every time there is a queue of tasks in front of a machine and the priorities are unclear, we have resource contention; a few tasks are fighting to be processed at the same time by the same resource."

"Exactly," I say. "And how do you handle such a case? You have learned that it is foolish to try to schedule the work of each and every machine. What are you supposed to do?"



"Identify the bottleneck," Charlie says.

"And then?"

"Then exploit it; schedule the sequence of work for the bottleneck."

"By that," I say, "you have eliminated any contention on the constraint. You don't ask it to do two things at the same time. And then? Charlie?"

"And then subordinate. Subordinate all the other resources to it."

"And what is the result?" I ask. I am getting good at asking rhetorical questions. Rhetorical to these students who have learned it from Johnny.

"By that," Charlie replies, "you have removed most of the overloads from the other resources. And the sporadic peaks of load that still exist can be absorbed by the buffers."

"Exactly," I say triumphantly. "Why don't we do the same in projects?"

"But in projects we don't have a bottleneck," Ted reacts too quickly.

"Really?" Mark asks ironically. "In your company you don't have a bottleneck? Remember, we are not talking about one project; we are talking about all of them together."

"I see."

"And what is the impact of not acknowledging such a bottleneck?" I ask. "It's not just that it creates havoc synchronizing between the projects. You will get the same devastating impact we get in production. Not paying special attention to the bottleneck, not protecting it from Murphy with buffers, unavoidably results in wasting time on the bottleneck."

"Which causes," Fred continues, "a reduction in the overall throughput of the organization. We deliver fewer projects in total than we could."

"So," Mark takes over, "we identified the bottleneck. Very easy, we knew it all along. It is our digital processing department. And then we scheduled their work."

"How?" Brian interrupts.

"The same way we do it in production. There the priority is mainly determined by the due dates of the orders. In our case, by the targeted completion dates of the projects."

Ruth continues, "And from there it was easy. We went back to dealing with each project as a stand-alone. The impact of the other projects was taken into account by the additional information we got from scheduling the digital processing department."

This explanation is too abbreviated for most.

"In every project we have steps done by digital processing," Fred elaborates. "The schedule of the digital processing gave specific start and finish dates for these steps. So, for each project we first did the work as if no other project existed. You know, removing any major contentions. Then we adjusted the project to fit the digital processing dictates."

"Did it change the critical chain?" Ted asks.

"For some projects it did," Fred acknowledges.

"Then you put in the buffers?" Brian checks.

"Of course," Mark answers. "But here comes a major point. All the buffers that we've talked about so far, project buffer, feeding buffers and resource buffers, are all protecting the individual project. Here we have to remember to also protect the bottleneck, the overall performance of our organization."

Ruth continues. "So we had to insert another buffer, the bottleneck-buffer. It's not as big a deal as it sounds. We decided that two weeks is more than enough, for us it is, and every path that goes through digital processing we schedule to start two weeks earlier. It's as simple as that."

The class is quiet. Everyone is trying to digest what they've heard. I don't break the silence.

"We are still not sure if scheduling just the digital processing department is enough," Fred says. "Remember, in production there is the need to sometimes consider not just the bottleneck but another capacity constraint resource, or two."

"How are you going to know?" Charlie inquires.

"We monitor the feeding buffers with hawk eyes, for early warnings," Fred answers. "If a resource contention starts to exhaust one feeding buffer after another, we'll know."



"But only then will we declare it as another resource constraint, not a minute before," Mark is fast to interject. "We are not going to be hysterical and consider every department a constraint just because they claim to be overloaded. We've learned our lesson. Never again are we going back to that nightmare."

We are sitting in a small deli in New York eating breakfast. Me and my Judith. When in Rome act like a Roman, so I've ordered a bagel with cream cheese and lox. Not bad.

New York is Judith's favorite hunting ground. She has perfected it to a science. We don't wander from shop to shop looking for something that will catch our eye. Oh, no, we are much more sophisticated. Judith plans it all in advance, down to the specific shops and the best routes.

Yesterday we were after an Oriental rug. After seven shops I was too exhausted to continue counting. At six o'clock we went back to the second place we visited in the morning and the battle started. Half an hour later, nine hundred and forty dollars poorer, we stepped out, the proud owners of a small, staggeringly beautiful, Persian rug. "We saved at least four hundred dollars," Judith summarized the day.

"What is our target today?" I ask my general.

"An antique coffee service."

"I thought you were pleased with the china you got last year?" I like it, even though I must admit that at the time I didn't like finding out we were another five hundred in the hole. I even made a small fuss.

"Our china is fine. But we need a silver set. For special occasions," and casually she adds, "you have cheese on your chin."

I almost choke. Special occasions, like twice a year when the queen of England pays us a visit.

"Wipe your chin," she reminds me.

I do. "What's the budget?" I dare to ask.

"No more than six thousand," she answers. "It won't be easy,

so this time you'll have to play your role when I'm bargaining. Don't just stand there like a depressed log."

Now that I have tenure we don't have to worry about saving for old age; the university pension will take care of that. But six thousand dollars? And for something that we'll never use? Then I see her point. It's a good idea.

Satisfied, I announce, "We are starting a collection."

"The Silver collection," Judith titles it.

I devour what's left of the bagels, and wave for a refill. "Over the years we'll build a collection that will be something." Jokingly I add, "Then we'll leave it to the town and people from all over will come to see it."

"I wish we could leave it to our kids," she says in a low voice.

"So do I. So do I."

Absentmindedly I drink my coffee. "I wish there was something we could do." I put the cup back. "Come on, Judith, let's conquer the city." I stand up, all ready to go.

"There is something," she says.

I move behind her to pull her chair back.

She doesn't stand up. "There is a way we can have a child."

I sit down again. "But, darling, I thought you were not willing to adopt?"

She puts her hand on mine. "There is a way to have our own children."

"But . . ." I feel weak.

She strokes my arm. "Nothing is wrong with my eggs or with your sperm."

"You mean a surrogate mother?"

"Yes."

I try to digest. We can have a baby.

"Rick . . . darling . . . let's go." She stands up. "It's out of our reach."

"No." I grab her hand and pull her back. "You're sure it's feasible? I know that in this thing nothing is guaranteed, but is it feasible?"



Standing, she replies, "Yes. But what's the point in torturing ourselves. We'll never be able to afford it. Let's go, honey."

"Are you willing to go through all it takes?"

"You know I am." She sits down.

"And if at the end we fail. Can you take it?"

"I'll still have you."

"Judith, if you are ready, I'll find the money. Whatever we need."

"There is one problem," she says.

"What?"

"We'll have to postpone the start of our silver collection. Is that okay with you?"

If it wasn't New York, we would probably be arrested for making a scene in public.