SCHEDULING TECHNIQUES IN
THE
REQUEST ORIENTED SCHEDULING ENGINE
(ROSE)

December 13, 1990

Agenda

• Introduction to ROSE
• NCC-ROSE (test results)
• ROSE Scheduling Approach
• Scheduling Techniques
• Summary

GSFC Contacts for ROSE Projects:

G. Mike Tong (Code 520) ATR for ROSE*Ada
Larry Hull (Code 520)
Nancy Goodman (Code 520) ATR for NCC-ROSE
Sylvia Sheppard (Code 520)
**ROSE Summary**

- **ROSE** is a prototype scheduling tool that has demonstrated viable solutions to difficult scheduling issues such as:
  - Fast, automated, conflict-free schedule creation (> 4,000 request/hour @ 2,000 req's.)
  - Schedule enhancement through post-processing: Best First Search for Schedule Enhancement (BFSSE).
  - Rescheduling / contingency scheduling techniques
  - Operator tools for computer-assisted scheduling (graphical interfaces, etc.)

- The **ROSE** effort involves the cooperation of experienced users, operators, and implementors of spacecraft data systems

- The **ROSE** effort has had positive impacts far beyond its original scope

---

**ROSE History**

- 1987 (FORD) Telecommunication Implications on Ground Systems (TIOS)
- 1988 (SEAS / Loral) Integrated Resource Scheduling Task (Distributed Scheduling)
- 1988 (SEAS / Loral) HCC Scheduling Prototype
- 1989 (SEAS / Loral) Comparison Study Report Released
- 1990 (SEAS / Loral) Continued Enhancement
NCC - ROSE Task Goals

- Prototype a viable generic NCC request scheduling process with predicted load levels for the 1995 timeframe using:
  - Existing ROSE prototype
  - Different request selection and placement strategies
  - Different scheduling algorithms

- Use requests that represent a realistic contention for TDRSS resources with realistic view periods

- Prototype required user request flexibility

- Evaluate FERN language for use in the NCC environment

- Determine tradeoffs between success rates and time-to-schedule for different scheduling algorithms

Accomplishments

- Verified ability of FERN to represent realistic generic requests by building and scheduling generic requests:
  - 31 Generic user requests
  - 11 Missions
  - Requests for 1645 activities per week
  - Realistic TDRS view periods
  - Realistic resource contention

- Prototyped and compared scheduling architectures
- Results documented in Scheduling Results Analysis Report for the NCC Prototype

- Able to schedule over 94% of anticipated requests for week long schedule in 1995 in less than 2 hours
Scheduling Concerns

- Need to reduce time needed to get a conflict-free schedule
- Satisfy customers
- Respond quickly to changes (Targets of opportunity, shuttle slips)
- Implement NASA policy
Current Approach

1. Users submit requests for services at a specific time.

2. INITIAL SCHEDULING (2 hours) - An initial schedule is created by computer.

3. CONFLICT RESOLUTION (3 to 5 days) - Operators phone users and ask:
   - What is the type of event? (orbit adjust, tape dump, etc.)
   - Can request be shortened?
   - Can request be moved?
   - Can request use a downgraded service (MA vs. SA)?
   - Can request use the other TDRS?
   - If neither conflicting user is flexible, choose the higher priority one.

4. Operators schedule PM and tests (hardware/software upgrades) around user requests.

5. If there is a conflict with a user, do the conflict resolution process.

SEAS
Systems, Engineering, and Analysis Support

ROSE Approach

1. Users and Operators submit flexible requests with preferences, constraints, and alternatives.

2. INITIAL SCHEDULING (1 to 2 hours) - An initial schedule is created (without conflicts).
   Some requested events are not scheduled.

3. CONFLICT RESOLUTION (2 to 5 hours) - Algorithms that imitate the human conflict resolution process are executed to try to schedule the non-scheduled requests.

4. (done)

5. (done)

SEAS
Systems, Engineering, and Analysis Support
Initial Scheduling

1. Add to schedule, allocate resources, note success.
2. Pick next instance.
3. Determine activity, constraints, step durations (flexible).
4. Choose from flexibilities.
5. Pick start time.
6. Start times.
7. Try to remove conflict.
8. Unschedule if necessary, note failure.
9. Success?

BFSSE Overview

- Start with an initial conflict-free schedule and some unscheduled requests.
- Identify one unscheduled request that you would like to try to schedule.
- The algorithm executes the following three steps repeatedly as needed until either a solution is found or a timeout occurs.
  - SELECT
    Find places on the schedule where the request almost fits.
  - MOVE
    Determine what requests need to be moved to schedule the unscheduled request.
  - RESCHEDULE
    Repeat the SELECT and MOVE steps for all moved requests.
BFSSE Example

- Goal is to add request “A” to the existing schedule shown below

- Three potential times for request “A” are shown

```
  J  K  L  M  N
  A? A? A?
  J  K  F  N
  Mission 1
  Mission 2
  Mission 3
```

- It is known that “A” cannot be scheduled anywhere if the rest of the schedule remains as is.

BFSSE Example (cont’d)

```
A
  /\  100
 /   \
A / T1
   \ /  300
    \  
A / T2
     /  150
    /   
A / T3
     /  
    /   
F
 `--`
  20
  /  
F / T4
 /  
F / T5
```

- Heuristics for Picking Times
- Heuristics for Selecting which activities to move

SEAS Systems, Engineering, and Analysis Support
Summary

- ROSE has shown to be an effective scheduler for solving the types of scheduling problems faced by the NCC
- The ROSE approach fully supports the NCC operations scenario
- Conflict-free schedules can be created in 2 to 4 hours instead of 3 to 5 days.
- ROSE can create schedules quickly enough that alternative contingency schedules are possible
- The ROSE conflict resolution strategy utilizes flexibilities in user requests to reduce conflicts