

A Web-Based Monitoring System for Multidisciplinary Design Projects

**James L. Rogers, Andrea O. Salas, and Robert P. Weston
NASA Langley Research Center**

**Presented at the
7th AIAA/USAF/NASA/ISSMO Symposium on
Multidisciplinary Analysis and Optimization**

St. Louis, MO

September 2-4, 1998

Outline

- **Background for this project**
- **Test problem for developing web-based capabilities**
 - **Sequencing**
 - **Monitoring**
 - **Displaying**
 - **Controlling**
- **Implementation of capabilities within an existing framework**
- **Plans**

Framework Definition

A framework for multidisciplinary design optimization is defined as a hardware and software architecture that enables integration, execution, and communication among diverse disciplinary processes.

Context of This Activity

- 1992-97 In-house framework development called FIDO (Framework for Interdisciplinary Design Optimization)**
- 1996-97 In-house evaluation of off-the-shelf commercial frameworks for purposes of acquiring an existing framework to replace in-house research framework (Salas presentation)**
- 1997 Begin migration to a commercial framework**
- Emergence of web technology**

Context of This Activity

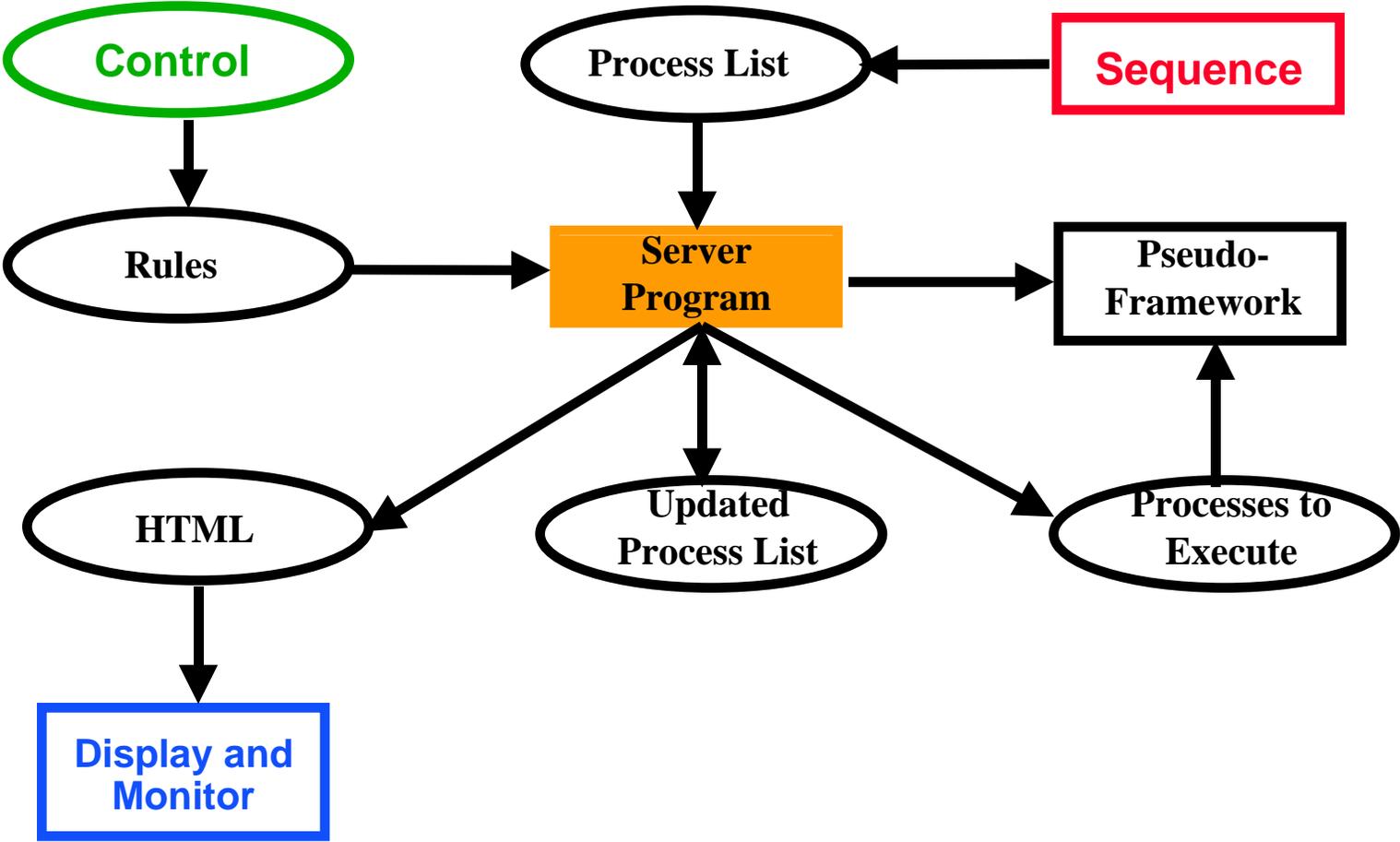
- **Evaluation uncovered apparent weaknesses in existing frameworks at that time:**

- **Sequencing processes**
- **Displaying results**
- **Monitoring project flow**
- **Controlling process flow**

- **Purposes of this project:**

- **Explore how process management & Web technologies, integrated into an existing framework, might improve areas of weakness**
- **No intent to build a new framework, only to demonstrate potential of process management & Web technologies to framework developers**

Monitoring System for Test Problem



Sequencing

Weakness

Existing frameworks typically

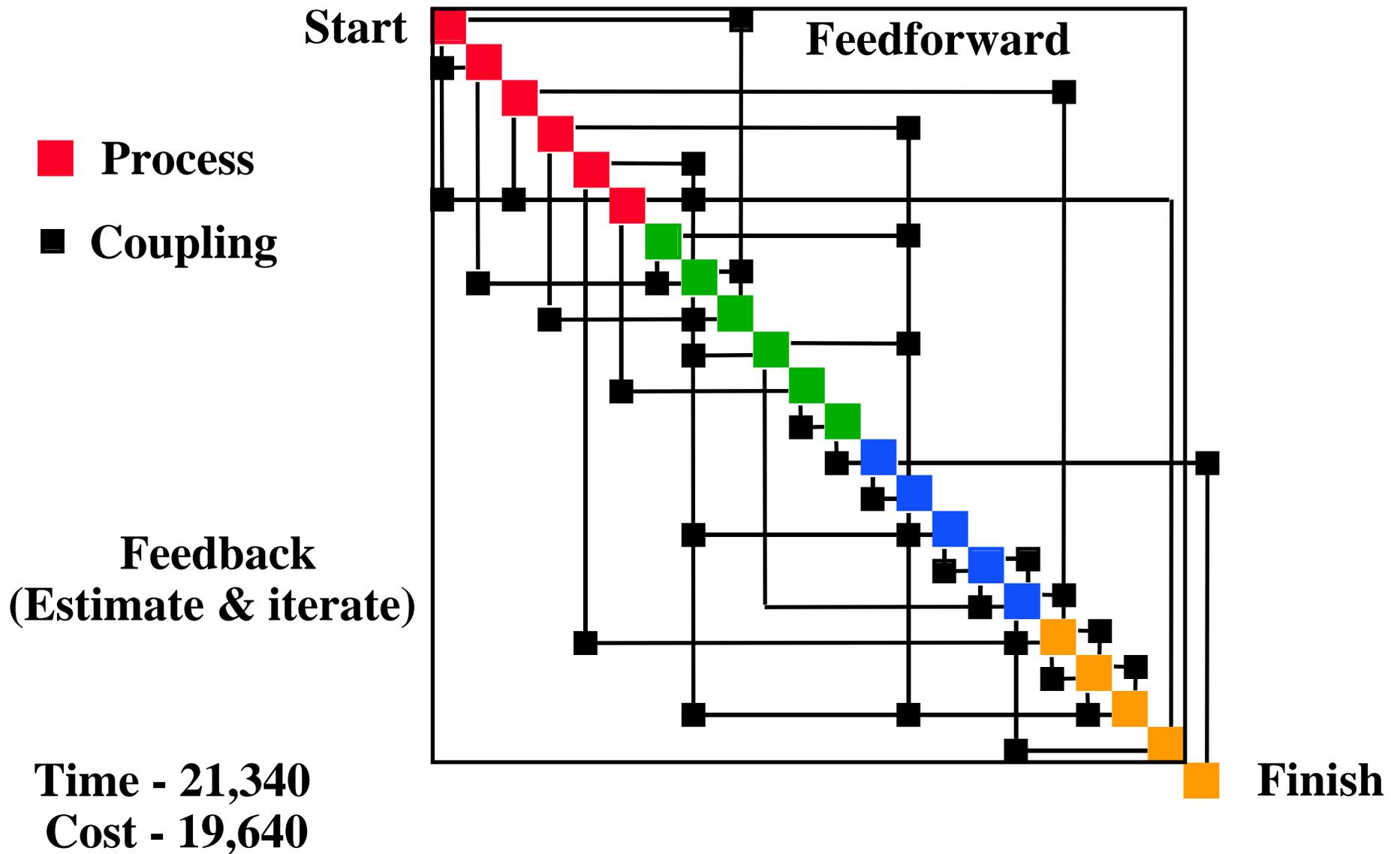
- did not optimize process sequence
- fixed the sequence to execute processes
- a fixed sequence is usually application dependent
- required significant labor to code sequence

Capability

For this approach, DeMAID (Design Manager's Aid for Intelligent Decomposition) applied as a pre-processor to determine optimum sequence

- Minimizes feedbacks
- Determines iterations
- Creates a list of processes and their couplings
- Makes process sequence application independent

Design Structure Matrix (DSM)



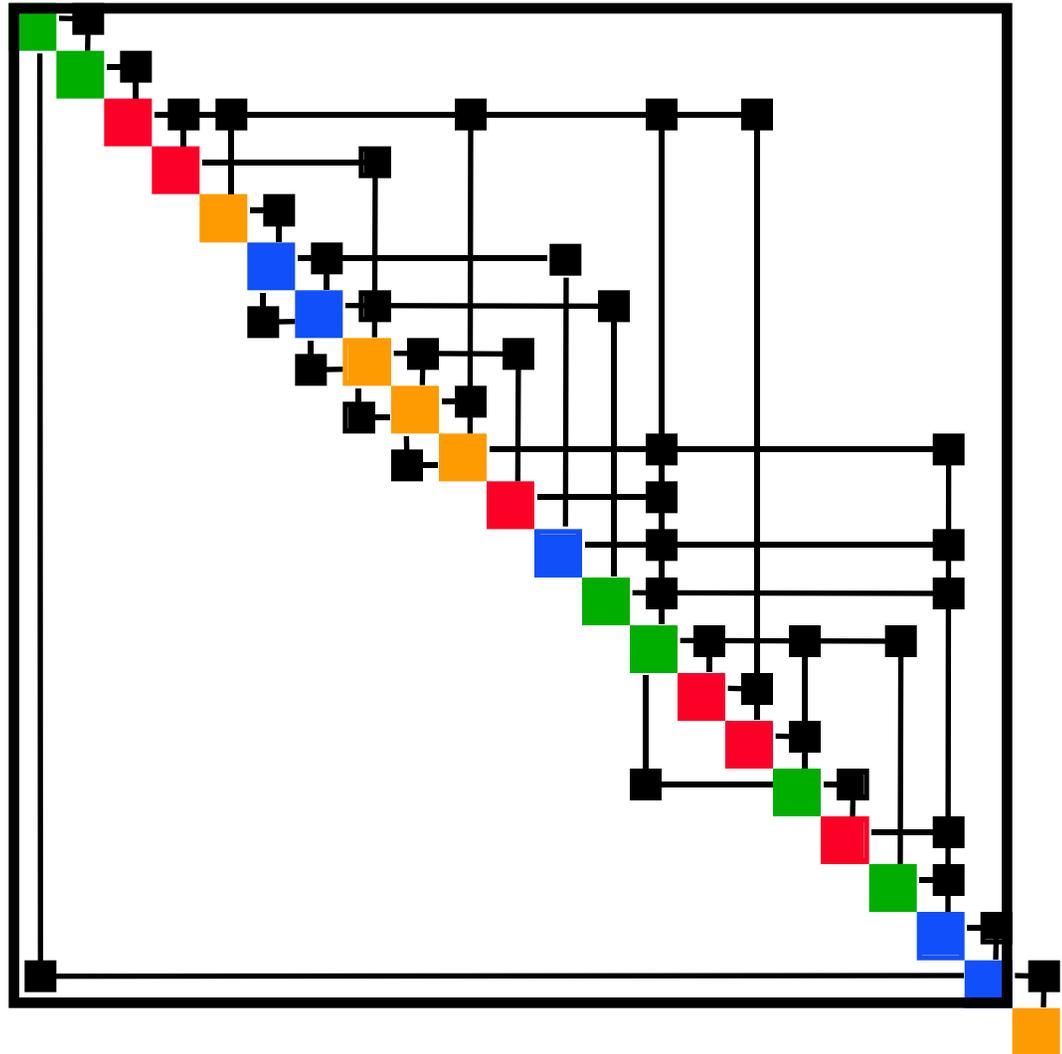
DSM for Optimized Sequence

Reductions

Time 21,340 to 3,800

Cost 19,640 to 3,220

**Also can determine
processes which can be
executed in parallel**



Server Program

The server program:

- reads the list of processes created by DeMAID
- read rules to determine which processes are ready for execution
- creates a file of processes ready for execution and calls the pseudo-framework
- updates the status for the list of processes file
- creates an HTML file to monitor the process flow on the Web in DSM format

Monitoring

Weakness

Existing frameworks did not allow project engineers to monitor the process flow.

Capability

- **The DSM in HTML format used for monitoring process flow**
- **Colors indicate the status of both processes and coupling data**
- **Links to process web pages provide access to more detailed information and data for each process**
- **The DSM is automatically updated every few seconds to provide insight to the most current status**

DSM for Test Problem in HTML Format

Links

Process Name	Process #	1	2	3	4	5	6	7	8	9
<u>init</u>	1	1								
<u>movehist</u>	2		2							
<u>anal1</u>	3			3						
<u>anal2</u>	4				4					
<u>step</u>	5					5				
<u>deriv1</u>	6						6			
<u>deriv2</u>	7							7		
<u>gsenext</u>	8								8	
<u>optimizer</u>	9									9

Processes - Diagonal

Turquoise - will not execute again

Yellow - waiting for data

Green - ready for execution

Black - removed from sequence

Coupling data - Off-diagonal

Blue - data is available

Red - data is not available

Gray - part of a circuit

Displaying

Weakness

Existing frameworks did not allow easy access to data displays from different computers and geographic locations.

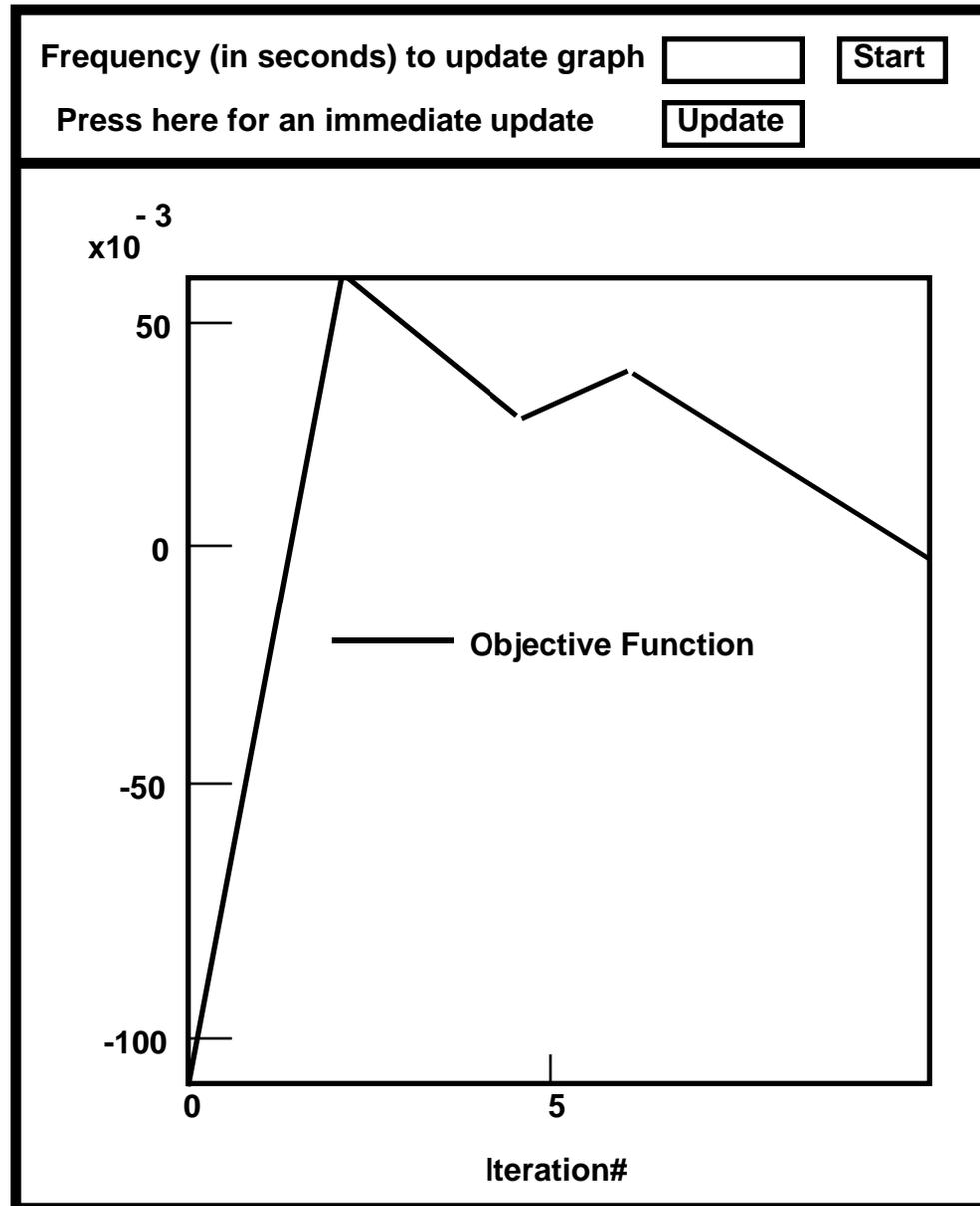
Capability

Java applets implemented to display both current and historical data and are invoked from the process web pages.

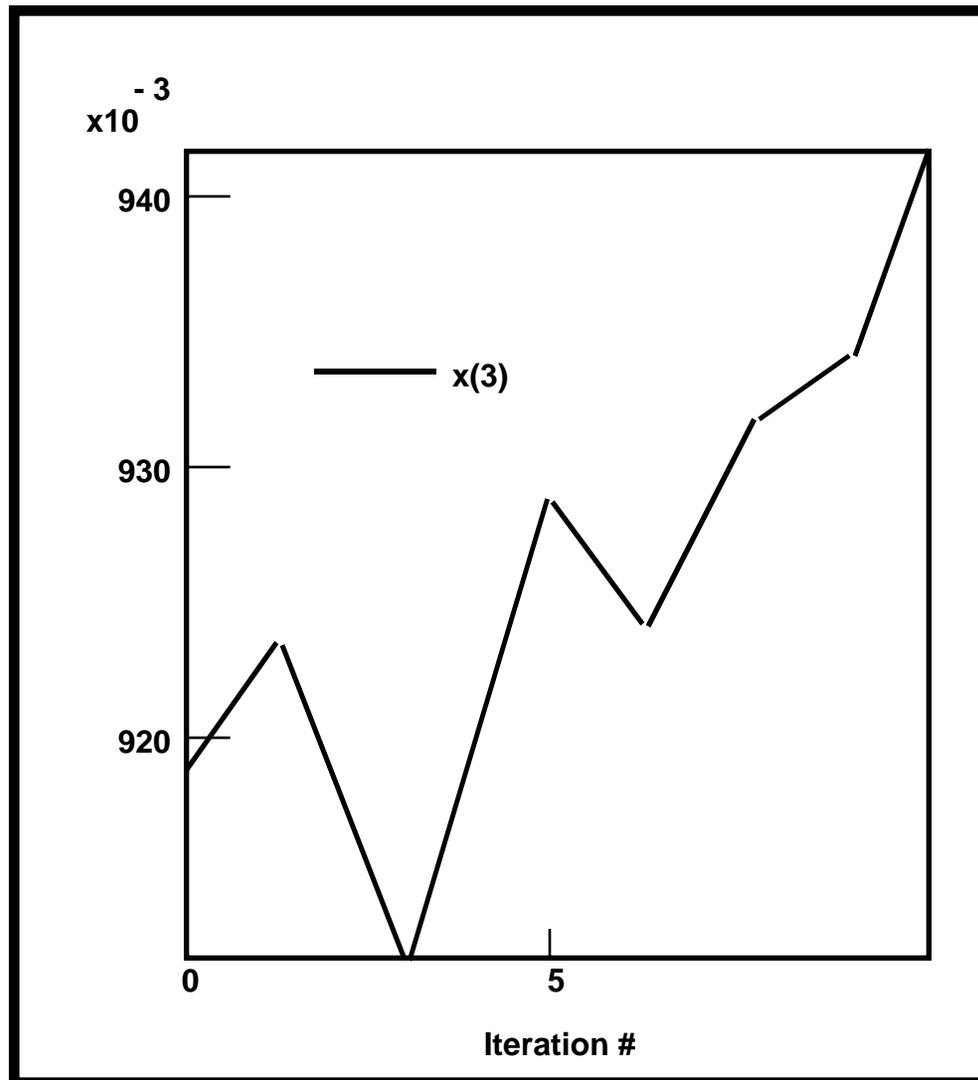
- Objective Function
- Design Variables
- Constraints Status

All of this information is readily available to anyone with access privileges to the web page.

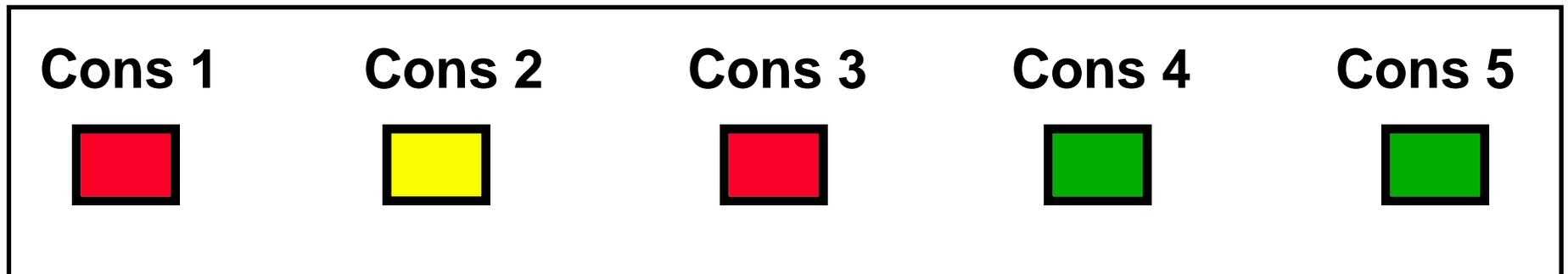
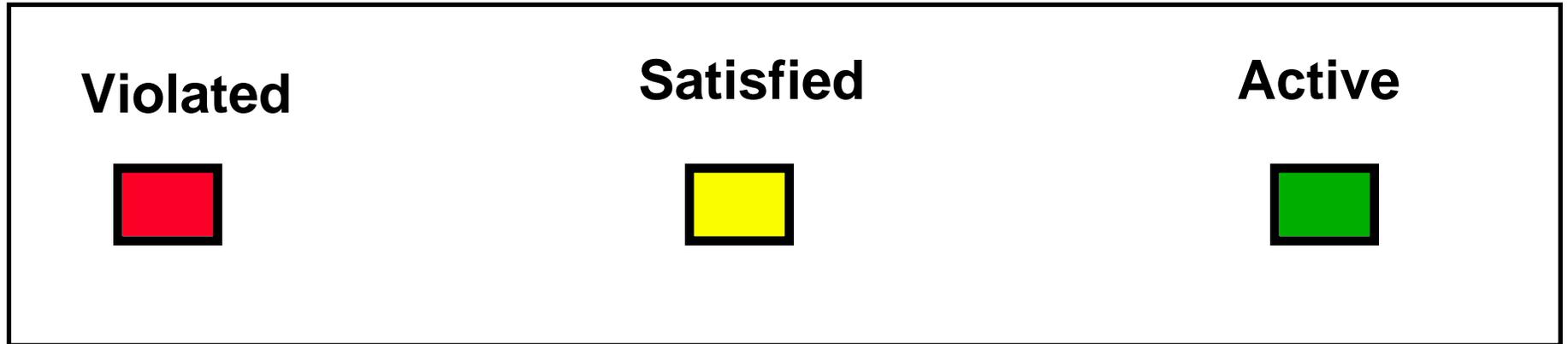
Test Problem Objective Function History



Test Problem Design Variable History



Constraint Display



Controlling

Weakness

Existing frameworks had a fixed process sequence and could not change once process began nor could the process be changed.

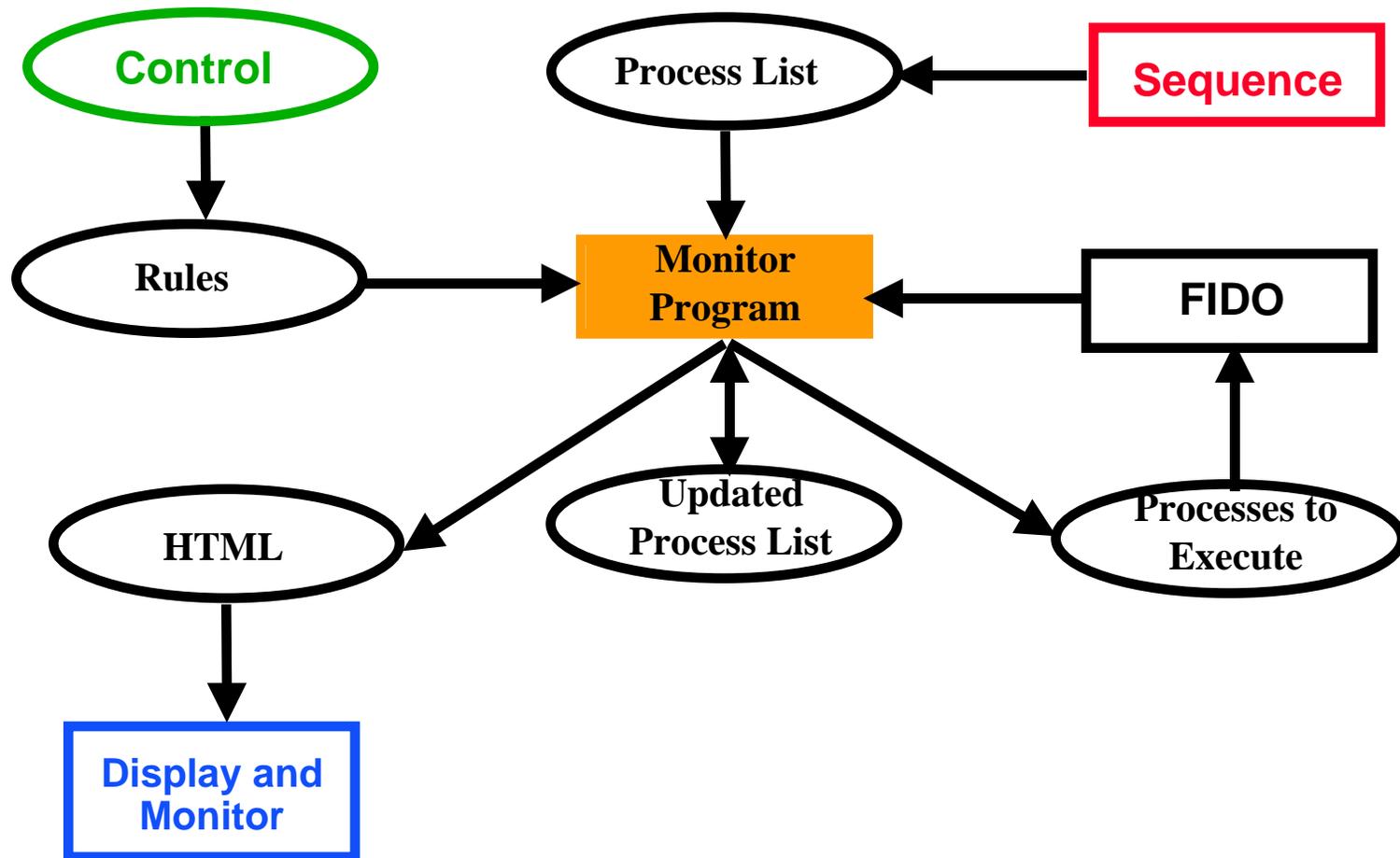
Capability

- Rules used to determine processes ready for execution.
- Control options set with HTML forms and passed to the server program.
 - Deactivate a process if the output from the process is not changing. The process may later be reactivated.
 - Select from among several programs for the same process. For example, select an approximation or a full analysis.

Implementation with an Existing Framework

- **Framework for Interdisciplinary Design Optimization (FIDO) was selected**
- **FIDO process calls are fixed**
- **Original implementation of a server program controlling and calling a framework not feasible**
- **FIDO modified to call monitor program**
- **FIDO processes blocked as “if” statements to reflect process execution**
- **Removed iterations from FIDO**

Implementation with an Existing Framework

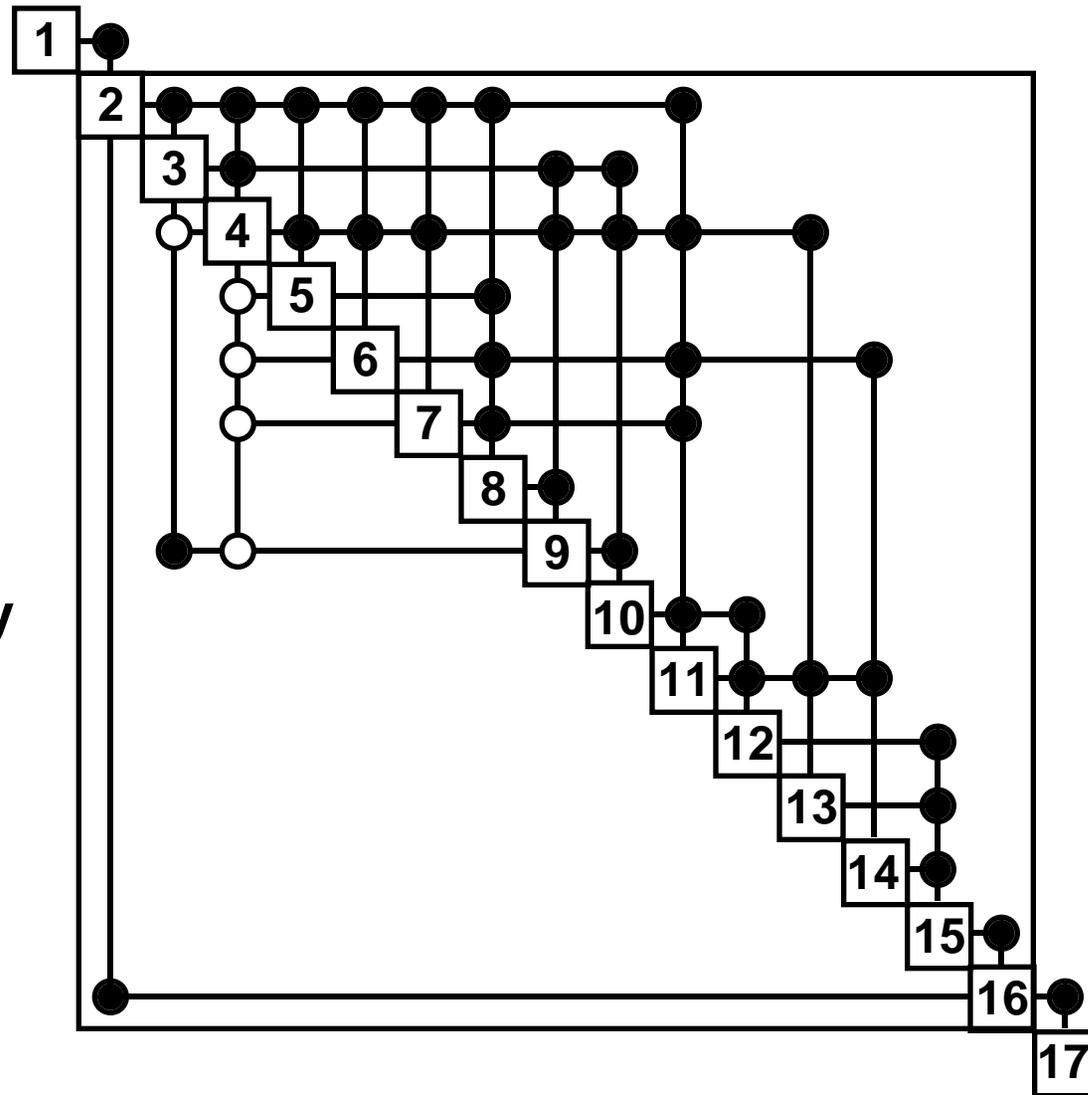


High Speed Civil Transport (2.1) Test Case

- More complex than test problem
- Implemented in FIDO in 1994
- Implemented in IMAGE (Hale, 1996) and iSIGHT (CSC, 1997)
- Contains low fidelity codes
- DeMAID used to create process sequence identical in sequence to current implementation in FIDO
- All but two feedbacks were deactivated to model the main iterations in the FIDO implementation
- User can specify number of iterations at each feedback

DSM for HSCT Problem

- | # | Name |
|----|--------------|
| 1 | Start |
| 2 | Waitstart |
| 3 | Calc WDD |
| 4 | Aero Analy |
| 5 | Calc CP |
| 6 | Perf Analy |
| 7 | Prop Analy |
| 8 | Sync Analy |
| 9 | Struct Init |
| 10 | Struct Analy |
| 11 | Opti Analy |
| 12 | Struct Grad |
| 13 | Aero Grad |
| 14 | Perf Grad |
| 15 | Sync Grad |
| 16 | Optimize |
| 17 | Exit |



Implementation Results

- **HSCT 2.1 problem tested through two optimization cycles of FIDO with with process calls working correctly**
- **FIDO development & maintenance stopped in 1997**
- **Change in OS and compiler (1997) created problems in FIDO that were deemed not worth resolving in view of transition plans**

Plans

- **Currently in a transition stage**
 - Updated version of commercial framework is being released 9/98 and will be evaluated
 - Interim in-house effort exploiting object-oriented implementation for HSCT design
- **Will consider resuming this project once framework transition is complete, especially since the DeMAID/*Web* code replaces code that is typically problem dependent and could possibly be a step towards making a framework more problem independent**
- **Additional capabilities to research**
 - Examine effects of different process sequences
 - Add capability to execute processes in parallel
 - Try approach on other frameworks