Outline

1. SNAP Goals & Constraints
2. SNAP Modeling
3. More on empirical studies
4. Camera Installation and running SNAP experiments
Goals

- **Squadron Capability**
  - Mix of quals: a matrix defining how many of which quals should be represented in the squadron

- **Average Pilot CRP (combat readiness %)**
  - CRP is a fractional value for mission codes
  - CRP is counted when pilot is current for that code

- **Pilot Competence**
  - A chart lists discrete codes a pilot should have current to be satisfactory

- **Equity in Pilot Hours**
  - E.g., within ~10% equity in a monthly basis
Basic Mission Resource Model

3 main mission types:
- FRAGS
- Builds
- Training
Training Codes Prerequisites

- Prerequisites are defined in the Pilot Training Manual (PTM):
  - For a given mission code the PTM specifies what are the prerequisite codes that a pilot must have current in order to fly the given code
  - Code currency is not constant and codes must be refreshed in order to remain current
  - PTM prerequisites are captured in xml form and is part of the SNAP code
Fly Day

The Fly Day is specified as a list of time intervals

- The sum of the intervals duration cannot exceed the maximum fly-day length constraint (typically of 10 hrs)

- Usually the Fly day consists of 1 or 2 intervals

- Can change daily or weekly

- The Fly day only limits the take-off and landing periods. Take-offs and landings cannot happen outside the Fly day
Crew Day

- The crew day is the period of time a pilot can be on base
  - Is a continuous interval typically < 12 hrs
  - Pilots cannot be scheduled to fly outside their crew day
  - There is a minimum distance between consecutive crew days that must be taken into account.
Day & Night Missions

- Day missions:
  - Take-off: Lower bound is 1/2 hour after sunrise
  - Landing: Upper bound is 1/2 hour before sunset

- Night missions:
  - "Night" is defined by the amount of light (LUX) and a given time interval after sunset
  - LUX and moon illumination and position constraints. For example a given mission might required to be performed 1hr after sunset and LUX < 0.002.
  - These rules as defined in PTM
  - Special rules for NS qualed vs Pilot In Training
Turn-arounds, Pits & Briefing Time

- Two main scheduling schemes:
  - Turns
  - Pits

- Pits are mostly a pilot constraint

- A typical daily schedule combines Pits and Turns

- Turns are mostly a maintenance constraint
Pits

**Pit Sequence**

- Pilots and aircrafts are the same
- In second leg can only fly certain missions codes restricted by the code of the first leg
- Time interval for which resources are needed:
  - Pilot: From beginning of briefing of first leg to end of debriefing of the last leg
  - Aircraft: From take-off of first leg to landing of last leg
  - Range: individual ranges are needed only during the "time in range" interval of each leg
- Planes have a 2hr turnaround time
- No restrictions between planes, pilots and codes after the Turn
- Time interval for which resources are needed:
  - Pilot: From beginning of briefing to end of debriefing
  - Aircraft: From take-off to landing. Aircrafts unavailable for a period of 2 hrs after landing
  - Range: during "time in range" interval
Pilot, Aircraft and Ranges Time

- Aircraft time sequence:
  - Time to range
  - Time in range
  - Return time from range
  - Turnaround time (for Turns only)

- Pilot time sequence:
  - Briefing time: 2 hours
  - Fly: as defined in DB (typically 1 hour)
  - Debriefing: 1 hour for training missions

- Ranges:
  - Time in range: typically $\frac{1}{2}$ hour
Schedule Types

- There are two basic Schedules types:
  - Normal Schedule:
    - The first take-off after the TURN must be at least 2 hours after the last landing of flights done before the TURN.
    - Not all take-off must be exactly 2 hrs after the last landing
  - Line Schedule:
    - Planes turnaround time don't match up
    - Planes are available for other missions after they have completed their 2hrs long turnaround time

- The schedule type is decided on a daily basis
- A Typical schedule is: 4-Pit-4 -TURN-4-Pit-4
Instructor Requirements and Formations

- Determined based on training code and pilot quals:
  - If the pilot is qualified for the code in the "Quals Needed to Fly Without IP" table then he can fly without and instructor
  - If NOT qualed then needs an instructor with the qual given in the "MissionSupportRoles" table.
  - Two main formations: Sections and Divisions
  - Section:
    - 2 planes. Lead and Wing.
    - Pilot Positions are: Section Lead and Wingman
  - Division:
    - 4 planes. Is formed by 2 Sections
    - Pilot positions are: Division Lead, Wingman to the division lead, Section Lead and Wingman for the Section Lead
  - Formation and the codes are in "MissionSupportRoles" table
Instructor Requirements and Formations II

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LATT</td>
<td>SLAT</td>
<td>221</td>
<td>LAT1955</td>
<td>Build</td>
<td>Wingman</td>
<td>222</td>
<td>223</td>
<td>224</td>
</tr>
<tr>
<td>LATT</td>
<td>LAT</td>
<td>222</td>
<td>LAT1955</td>
<td>Build</td>
<td>Wingman</td>
<td>222</td>
<td>223</td>
<td>224</td>
</tr>
<tr>
<td>LATT</td>
<td>LAT</td>
<td>223</td>
<td>LAT1955</td>
<td>Build</td>
<td>Wingman</td>
<td>222</td>
<td>223</td>
<td>224</td>
</tr>
<tr>
<td>LATT</td>
<td>LAT</td>
<td>224</td>
<td>LAT1955</td>
<td>Build</td>
<td>Wingman</td>
<td>222</td>
<td>223</td>
<td>224</td>
</tr>
<tr>
<td>LATT</td>
<td>LAT</td>
<td>225</td>
<td>LAT1955</td>
<td>Build</td>
<td>Wingman</td>
<td>222</td>
<td>223</td>
<td>224</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage</th>
<th>Flight Training Code</th>
<th>Pilot Qual</th>
<th>Leadership Qual</th>
</tr>
</thead>
<tbody>
<tr>
<td>LATT</td>
<td>SAG</td>
<td>214</td>
<td></td>
</tr>
<tr>
<td>LATT</td>
<td>AG</td>
<td>215</td>
<td></td>
</tr>
<tr>
<td>A/G</td>
<td>AG</td>
<td>216</td>
<td></td>
</tr>
<tr>
<td>A/G</td>
<td>AG</td>
<td>217</td>
<td></td>
</tr>
<tr>
<td>A/G</td>
<td>AG</td>
<td>218</td>
<td></td>
</tr>
<tr>
<td>A/G</td>
<td>AG</td>
<td>219</td>
<td></td>
</tr>
<tr>
<td>A/G</td>
<td>SLAT</td>
<td>220</td>
<td></td>
</tr>
<tr>
<td>A/G</td>
<td>SLAT</td>
<td>221</td>
<td></td>
</tr>
<tr>
<td>A/G</td>
<td>LAT</td>
<td>222</td>
<td>912</td>
</tr>
<tr>
<td>A/G</td>
<td>LAT</td>
<td>223</td>
<td>912</td>
</tr>
<tr>
<td>A/G</td>
<td>LAT</td>
<td>224</td>
<td>912</td>
</tr>
<tr>
<td>A/G</td>
<td>LAT</td>
<td>225</td>
<td>912</td>
</tr>
</tbody>
</table>
Range Capabilities

Range selection based on
- Availability
- Training code: suitability and preference:
  • Each range has a list of mission codes it supports
  • A measure of how suitable it is for that code in a scale from 0 to 5
- Range are typically available in 1/2 hr slots
- Support merging and splitting of ranges defined on a per range basis (underutilized until guidance interface supports it):
  - Altitude-based splitting/merging
  - Geography-based splitting/merging
Model: Basic Idea

Approach A
(ISI and Cornell start)

Set of Constraints
BIG CSP

SNAP Sortie Controller:
“SC give me the N next missions”

m₁ to mₙ
CSP

N missions graph

Improve Solution/Complexity Analysis/Optimization

Optimal Solution/Complexity Analysis
N missions problem: FRAG Missions

FRAG Missions problem:

- We have $N$ missions $m_i$ $i=1,2,…N$
- Each mission has $N_r$ roles
- *The time for the mission is predefined*
- To schedule the mission at the given time-slot $TS$ we need the following resources:
  - A qualified range available at time slot $TS$
  - A qualified pilot for each of the $N_r$ roles in the mission
  - A qualified aircraft for each of the $N_r$ roles in the mission
FRAGS Resource Sets

- $(1 + 2N_r)$ dimensions for each mission
- For each of these dimensions define the following sets of eligible resources:
  - Eligible means: qualified + available at TS
  - Ranges set with all eligible “range-slots” at $R = \{R_1, R_2, \ldots, R_4, \ldots\}$
  - Pilots set with all eligible “pilot-slots” to fly mission $M_i$ in role $r \rightarrow P_r$
  - Aircrafts set with all eligible “aircrafts-slots” to fly mission $M_i$ in role $r \rightarrow A_r$
- Graph: Missions are nodes. Edge connected if they overlap in time
N missions problem: Training Missions

- Training Missions problem:
  - We have $N$ missions $m_i$, $i=1,2,…N$
  - Each mission is to train a given pilot $P_T$ to fly a training code $C_m$
  - To schedule the mission at time-slot $TS$ we need the following resources:
    - A qualified range available at time slot $TS$
    - A qualified pilot for each of the $N_r$ roles in the mission (exclude pilot $P_T$ from this list)
    - A qualified aircraft for each of the $N_r$ roles in the mission (exclude aircraft for pilot $P_T$ from this list)
    - A qualified aircraft for pilot $P_T$
Training Missions Resource Sets

- 2 \((1+N_r)\) dimensions for each mission
- For each of these dimensions define the following sets of eligible resources:
  - Eligible means: qualified + available
  - Ranges set with all eligible “range-slots” \(R = \{R_1(t_1), R_1(t_3),\ldots, R_4(t_3),\ldots\}\)
  - Pilots set with all eligible “pilot-slots” to fly mission \(M_i\) in role \(r\) \(\rightarrow P_r\)
  - Aircrafts set with all eligible “aircrafts-slots” to fly mission \(M_i\) in role \(r\) \(\rightarrow A_r\)
  - Aircrafts set with all eligible “aircrafts-slots” for pilot \(P_T\) to fly mission \(M_i\) \(\rightarrow A_T\)

- Graph: ?
Training Missions Graph: Effective Colors (I)

- Group resource-slots into *effective colors* or eligible *states* that a given mission may assume.
Training Missions Graph: Effective Colors (II)

Finding the set of possible colors is already complex. How can we do it efficiently?
- E.g.: Random search until reach cutoff

Graph:
- Missions are nodes and effective colors are available colors and edge connect them if some colors overlap (over-constraining)
- Effective colors are nodes and edge connect them if they share some resource. Missions where that color appear are the possible ways of painting the node
- Other possible SAT encoding
Training Missions Graph (II)

- Other approaches?
- Optimization?
Resource Availability Ratios

■ Idea: Identify resource availability levels at successful attempts and learn why missions fail

  – When mission announces that it’s trying a new time-slot:
    • Count the number of aircrafts, pilots and ranges currently competing at that time-slot
    • Count the number of aircrafts, pilots and ranges currently available at that time slot
    • Compute aircraft ratio, pilot ratio and range ratio as ratio = available/competing
    • Take into account pilot time, aircraft time and range time

■ Future: XSLT implementation in CAMERA
Resource Availability Ratios: Results (I)
Pilot Availability Ratios

![Pilot Availability Ratios Chart]

- **Unsatisfied**
  - Pilot Ratio 1: 0.02%
  - Pilot Ratio 2: 0.04%
  - Pilot Ratio 3: 0.05%
  - Pilot Ratio 4: 0.28%
  - Pilot Ratio 5: 0.24%
  - Pilot Ratio 6: 0.5%
  - Pilot Ratio 7: 0.14%

- **Satisfied**
Aircraft Availability Ratios

![Bar chart showing aircraft availability ratios for different aircraft ratios. The chart displays the percentage of unsatisfied and satisfied aircraft for each ratio, ranging from 0.005% to 0.5%.](image-url)
Range Availability Ratios
CAMERA Code Installation

Installation instructions are in Martin Frank’s homepage: www.isi.edu/~frank/dceinstallation.html

- Cygwin
- Emacs
- CVS
- Environment variables
- CVS login
- Checkout source code
- cd to na level and compile with “make cio”
- cd to snap/problemgen run .sh file
ISI Team

ATTEND: (www.isi.edu/attend)
  Robert Neches
  Alejandro Bugacov
  Baoshi Yan
  Donghan Kim

CAMERA: (www.isi.edu/camera)
  Robert Neches
  Pedro Szekely
  Martin Frank
  David Benjamin
  Jinbo
End
Instrumentation tools

- It’s all about counting and downcasting messages.
- Coding **listeners** one can implement actions at the sending (or receiving) time of a message.
- Downcast a message to determine an action based on the message type.
- Starfields and scaling measurements are implemented using this mechanism.
Example: Measuring time at which a mission gets filled

```java
public void announceMessageAboutToBeSent(
    EaApiParticipant to,
    EaApiParticipant from,
    EaApiParticipant replyTo,
    EaApiMessage x)
{

    if(x instanceof EaApiNegotiationMessage) {
        EaApiNegotiationMessage m=(EaApiNegotiationMessage)x;

        boolean debugAE=true;
        // A sortie announced that it's trying a new time.
        if(m instanceof PrApiRcRequest) {
            PrApiRcRequest r=(PrApiRcRequest)m;
            if(r.getUseMainRequest() instanceof SnPrSortieTryingANewTimeSlotRequest) {
                SnPrSortieTryingANewTimeSlotRequest req=(SnPrSortieTryingANewTimeSlotRequest)r.getDomainRequest();
                SnPrSortieTryingANewTimeSlotResource n Req=try���eutingANewTimeSlotResource(req);
                TimeInterval newTime = a.getNewAttempt();
                String id = a.getSortieId();
                computeResourcesAtNewSortieSlot(newTime, id);
            }
        }

        if(m instanceof PrApiCoCommitment) {
            PrApiCoCommitment c=(PrApiCoCommitment)m;
            EaApiDomainResource d=c.getDomainResource();
            if((d instanceof SnApiSortieResource)) {
                String id=d.getSenderId();

                long la = System.currentTimeMillis();
                sNumber++;

                if(p == null){
                    p = SnExScalingMain.openFileWriter("schedRate.csv");
                    missionsList = new ArrayList();
                }

                if(SnExScalingMain.runNumber == 1){
                    missionsList.add(new long[]{SnExScalingMain.maxRunNumber+1});
                }

                --\** SnExExampleListner.endCode **--

    Auto-saving... done
```
Measurements

- During a run
  - $N(t)$: Number of filled missions at time at which a new mission is filled

- At the end of each trial run
  - Total number of generated missions
  - Resource availability
  - Fraction filled
  - Execution time

- Over several runs
  - Averages of $N(t)$ over several runs
Resource Availability: Density Control

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

80%  
50%  

range blocking interval  
pilot blocking interval
CAMERA-SNAP Experiments Results

Experimental procedure:

1. The number of Pilots, Ranges and Aircrafts was kept fixed and their percentage availability throughout the Planning Horizon was varied randomly (to meet the input availability density) by blocking time intervals of their available time.

2. The size of the problem was controlled by randomly varying the total number of planned missions during the planning horizon (for FRAGS).

3. The planning horizon was kept fixed at 1 week.

4. The execution time was measured in batch mode without any GUI cycles taken into account.

5. FRAGS only and FRAGS/Training Mix.
Experimental Setup

Run in batch mode with a bash shell script file

```java
java -mx128m -ms64m -Ddce.install.path=${DCE_HOME_PCSYNTAX} -Dresource.subdir=`cygpath -w \`pwd\`` edu isi dce na snap executive SnExScalingMain -simulation.trace.output.filename.head xxx-$$ -batch true -randomNumActivitiesWindow 25 -createObserver false -executionPolicy random -displayGui true -displayCommitments true -displaySchedules true -scheduleResolution 30 -displayQuals true -displayMessages false -displayRisk true -displayComputation false -showSortiesGraph true -showCrpGraph true -showNumScheduledActionsGraph false -showPendingActivitiesGraph true -showNumMessagesGraph true -showCommitRatioGraph true -pauseAfterInitialization true -pilotTrainingManualFilename ../pilotmodel/PilotTrainingManual-AV8PQC-Draft.xml -rangeQualsFilename allKnownRanges.xml -rangeAssignmentsFilename rangesvma513ForReal.xml -eventsFilename eventsvma513.xml -aircraftFilename aircraftvma513ForReal.xml -pilotsFilename pilotsvma513ForReal.xml -startTime 2000-07-26T09:30:00Z -sortiesFilename sortiesvma513ForReal.xml -sortieCrawlingDuration 30 -useEquityBasedVersion false -pilotRfcDelay 0 -setupFilename setupForRealAB.csv -pilotRenegesForABetterOffer false $*
```

Pilots, Aircrafts and Ranges initial availability data

Setup File
Participants (Agents)

- Sortie Controller
- Pilot Market
- Range Czar
- Aircraft Market
- Pilots
- Aircrafts
- Ranges
Commit-Decommit Protocol

Request for commitment
Commitment
Decline
Decommitment
Renege

Agent 1
Agent 2

Process alive
Process stops here