

A Framework for Analyzing and Designing Systematic Yet Flexible Systems

Todd R. Johnson & Eliz Markowitz



Previous Work on Systematic Yet Flexible (SYF) Architecture

- Perer & Shneiderman's Seven SYF Design Goals
 - See an overview of the sequential process of actions
 - Step through actions
 - Select actions in any order
 - See completed and remaining actions
 - Annotate their actions
 - Share progress with other users
 - Reapply past paths of exploration on new data

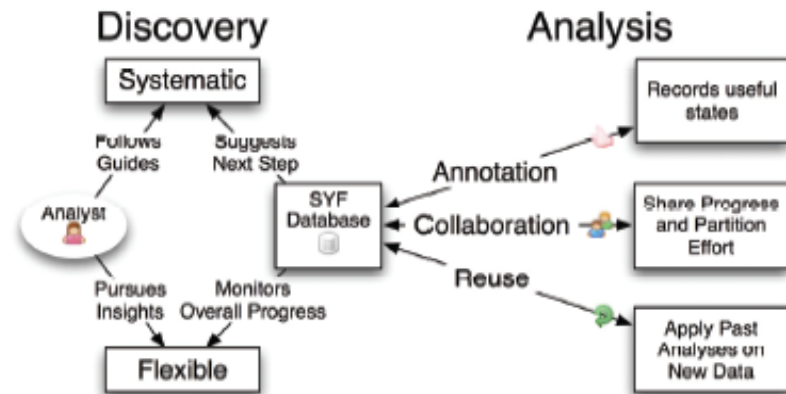


Figure 1. The SYF infrastructure facilitates discovery by providing systematic guides while also allowing users to flexibly pursue insights. SYF also facilitates analysis by allowing users to easily annotate during exploration, share exploration results with colleagues and partition effort, and reapply past exploration paths on new data sets.

Perer A, Shneiderman B. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. *Proceedings of the 13th International Conference on Intelligent User Interfaces*; 2008; New York, NY. 109-118.

Systematicity & Flexibility

- Systematicity
 - Provides structure to ensure consistency, efficiency, and safety by imposing necessary structure
- Flexibility
 - The ability to constantly adapt to circumstances and still reach the goal state
- Systematicity & Flexibility are at odds with one another

Systematicity & Flexibility in Healthcare

- There is a need for improved quality and efficiency in healthcare
- Has led to the introduction of consistent, systematic approaches to improve efficiency, safety, and effectiveness.
 - Standard operating procedures
 - Clinical Guidelines
 - Decision support
- But flexibility is needed to accommodate variation found in the healthcare field

Problems with Current Methods

- Too much systematicity
 - Decreases quality and efficiency
 - Lead to caregiver resistance
 - “Creative” Workarounds
 - Typing “Diabeetes” to avoid triggering automatic decision support
- Too much flexibility
 - Decreases quality
- Need for a SYF system
 - Supports graceful degradation from idealized practices to those that are more suitable for a given situation

Goals of an SYF Design Framework

- Guide the design of systems that support graceful degradation from idealized practices to those that are more suitable for a given situation
- Allow exploration of trade-offs among designs
- Provide an objective measure of flexibility for comparing designs

The Framework

- Identify a task (a problem to be solved)
- Analyze three problem spaces
 - The problem space represents all possible states and actions for a problem
 - Idealized space, Natural space, & System space
- Identify the components of a problem space
 - Symbolic Representation of state
 - Set of operators (the actions)
 - Initial State
 - Goal State(s)

Problem Spaces

- Natural Space
 - Identifies the task actions and the associated natural constraints
- Idealized Space
 - The best practice
- System Space
 - Specifies how the task is done in the system
 - Identifies both deviations from the idealized space and how the system supports and/or enforces the constraints in the idealized space

Let's Consider a Task...

- Central Venous Catheter Insertion
 - Used to deliver medications and/or fluids
- Approximate Order of Actions
 - Sterilize Site
 - Drape patient
 - Put Hat On
 - Put Mask On
 - Put Gown On
 - Wash hands
 - Glove up
 - Insert Central Line
 - Apply Sterile Dressing



State Representation & Operators

- Logical preconditions on the state and how the operators change the state

Operator	Precondition	Action
Sterilize Site	!sterilizedSite	sterilizedSite -> True
Drape patient	!drapePatient && sterilizedSite	drapePatient -> True
Put Hat On	!hatOn && drapePatient	hatOn -> True
Put Mask On	!maskOn && drapePatient	maskOn -> True
Put Gown On	!gownOn && hatOn && maskOn	gownOn -> True
Wash hands	!washedHands && gownOn	washedHands -> True
Glove up	!glovesOn && washedHands	glovesOn -> True
Apply Sterile Dressing	!sterileDressing && centralLineInserted	sterileDressing -> True
Insert Central Line	!centralLineInserted && glovesOn	centralLineInserted -> True

INITIAL STATE

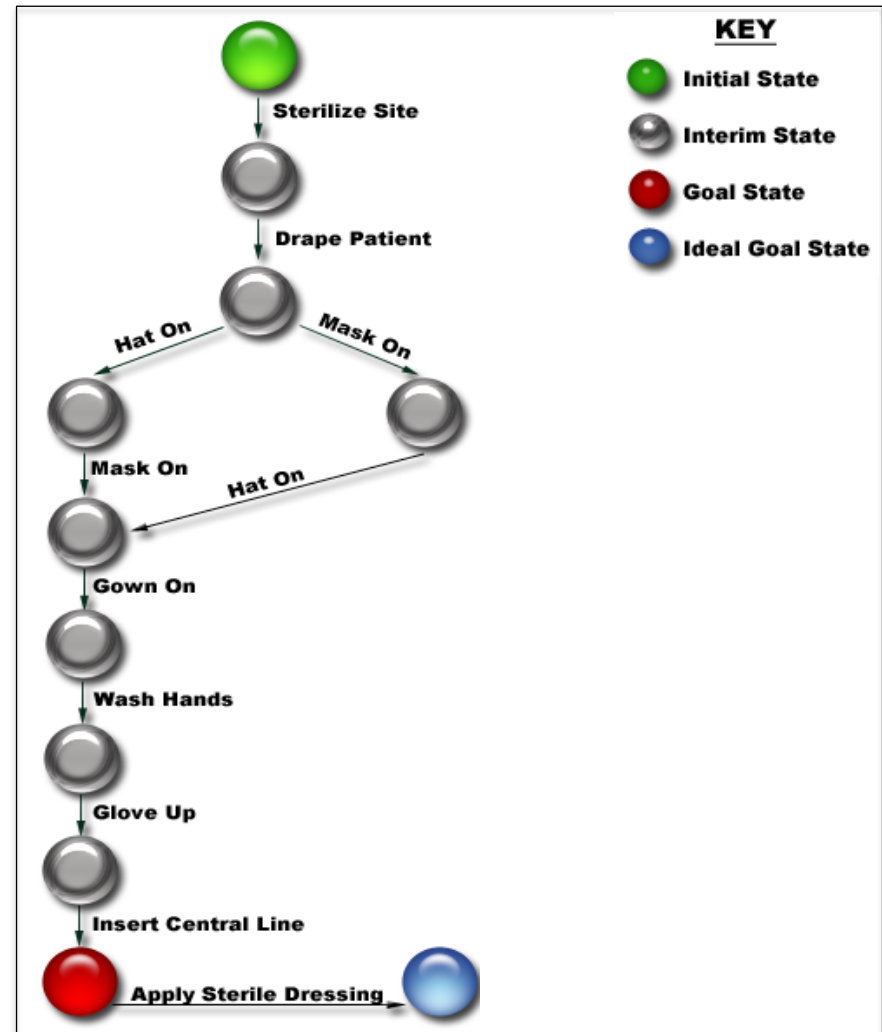
- centralLineInserted -> False
drapePatient -> False
glovesOn -> False
gownOn -> False
hatOn -> False
maskOn -> False
sterileDressing -> False
sterilizedSite -> False
washedHands -> False

GOAL STATE

- centralLineInserted -> True
drapePatient -> True
glovesOn -> True
gownOn -> True
hatOn -> True
maskOn -> True
sterileDressing -> True
sterilizedSite -> True
washedHands -> True

The Idealized Space

- Best specified as a work domain ontology (WDO) for the task.
- **WDO**- Defines the explicit, abstract, implementation-independent description of a task
- **Assumptions**
 - 1) A single caregiver accomplishes the entire task
 - 2) All supplies needed to do the task are available
 - 3) There is sufficient time to accomplish the procedure according to best practices
- The assumptions allow us to better assess the validity and score of the idealized space



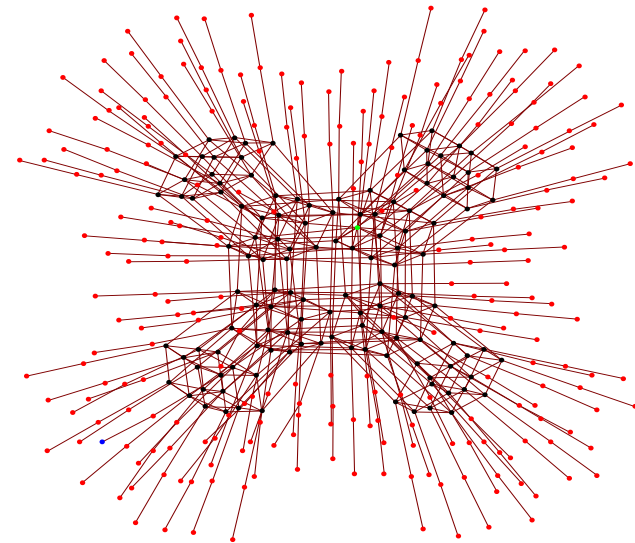
The Natural Space

- Assumptions, State Representation, and Operators are identical to the Idealized Space
 - Preconditions reflect hard constraints found in the task environment
- The Initial State is identical to the idealized space
- The Goal State is any state in which the central line is placed

Operator	Precondition	Action
Sterilize Site	!sterilizedSite && !centralLineInserted	sterilizedSite -> True
Drape patient	!drapePatient && !centralLineInserted	drapePatient -> True
Put Hat On	!hatOn && !gownOn && !centralLineInserted	hatOn -> True
Put Mask On	!maskOn && !gownOn && !centralLineInserted	maskOn -> True
Put Gown On	!gownOn && !centralLineInserted	gownOn -> True
Wash hands	!washedHands && !centralLineInserted	washedHands -> True
Glove up	!glovesOn && !centralLineInserted	glovesOn -> True
Apply Sterile Dressing	!sterileDressing && centralLineInserted	sterileDressing -> True
Insert Central Line	!centralLineInserted	centralLineInserted -> True

Analysis of The Natural Space

- 384 total states
 - 286 goal states
- 13,004 paths to any state in which the central line is inserted
 - 1,680 possible paths to the “idealized” state
 - Only 2 paths contain the appropriate sequence of 9 actions



The network diagram shows that the natural space is more complex and has more apparent flexibility than the idealized space.

Analysis of the Natural Space

- Visibility of system state
 - Some actions make no visible change to the state
 - Washing hands, sterilizing site (unless a staining solution is used)
 - Can lead to errors of omission and commission
 - Insufficient information to detect idealized goal state (depends on path to the goal)
- Natural constraints on action sequence
 - Idealized sequence is not supported by natural constraints
 - Hat and mask cannot be put on once gown is on, but all other actions can be done at any time
- Chance of post-completion errors
 - Sterile pack is placed after completion of main goal

Berenholz & Checklists

Feature Article—Continuing Medical Education

Eliminating catheter-related bloodstream infections in the intensive care unit*

Sean M. Berenholtz, MD, MHS; Peter J. Pronovost, MD, PhD; Pamela A. Lipsett, MD; Deborah Hobson, BSN; Karen Earsing, RN, MS; Jason E. Farley, MSN, MPH, CRNP; Shelley Milanovich, RN, MSN, ACNP; Elizabeth Garrett-Mayer, PhD; Bradford D. Winters, MD, PhD; Haya R. Rubin, MD, PhD; Todd Dorman, MD; Trish M. Perl, MD, MSc

- Nurse observes procedure and fills out checklist
- If not an emergency, nurse is empowered to stop the procedure when it deviates from the guidelines
- Central line insertion cart is readily available and contains all supplies needed to comply with best practice

Catheter-related Blood Stream Infection Care Team Checklist			
Purpose:	To work as a team to decrease patient harm from catheter-related blood stream infections		
When:	During all central venous or central arterial line insertions or re-wires		
By whom:	Bedside nurse		
1. Today's date	____/____/____ month day year		
2. Procedure:	<input type="checkbox"/> New line	<input type="checkbox"/> Rewire	
3. Is the procedure:	<input type="checkbox"/> Elective	<input type="checkbox"/> Emergent	
4.	Yes	No	Don't know
Before the procedure, did the housestaff:			
Wash hands (chlorhexidine or soap) immediately prior	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sterilize procedure site	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Drape entire patient in a sterile fashion	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
During the procedure, did the housestaff:			
Use sterile gloves	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Use hat, mask and sterile gown	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Maintain a sterile field	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Did all personnel assisting with procedure follow the above precautions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
After the procedure:			
Was a sterile dressing applied to the site	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Please return completed form to the designated location in your ICU.			

System Space & Checklist Use

- System space:
 - Allows switching between idealized and natural space
 - Achieves flexibility by accepting a different goal and relaxing action constraints in an emergency
 - Encourages and Enforces idealized practice through cart, checklist, and external monitoring and intervention
 - Checklist increases visibility of system state

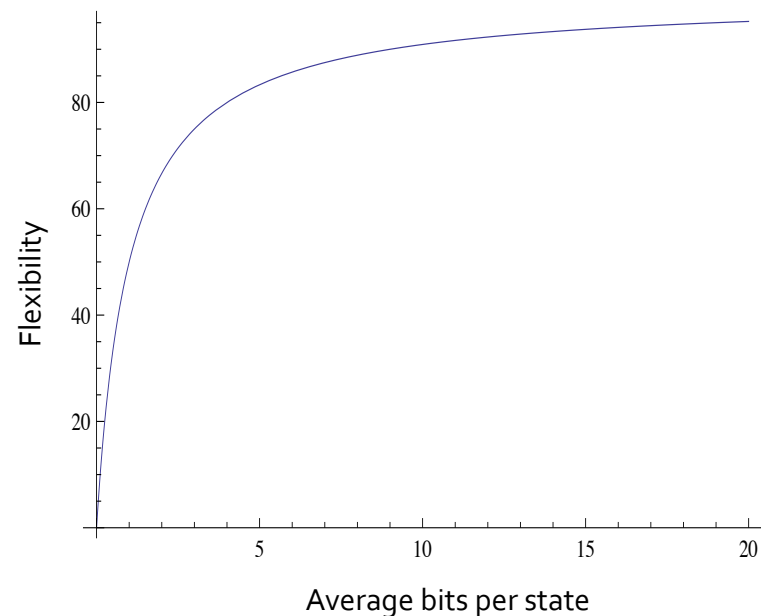
Measuring Flexibility

Common Sense View of Flexibility

- If there is only a single correct way to complete a task, that task has 0% flexibility.
- If any sequence of task actions leads to the goal, then the task has 100% flexibility.

Information Theoretic Measure of Flexibility for a Task Problem Space

- The average number of bits needed to select the next action for each non-terminal state in a problem space.
- We can convert bits, n , to percentage flexibility using the formula: $\frac{100n}{1+n}$



Idealized Space: Only One Action is Necessary

- Table A has ten blocks and Table B is empty. The goal is to place any one block from Table A onto Table B.
- There are *10* possible actions for the initial state, and any action leads to a goal state, resulting in *1* non-terminal state with *10* actions:
- (Sum of bits per state)/(# of states) for all non-terminal states
 - Average Bits per state: $\text{Log}_2(10)/1 = 3.32$ bits
 - Flexibility: 76.86%

Idealized Space: Order Does Not Matter

- There are ten blocks on Table A and the sequence in which you move the blocks to Table B does not matter as long as you move all ten blocks.
 - Once you have accomplished one sub-goal, the remaining sub-goals are constrained.
 - If there are n sub-goals, there are $n!$ possible paths.
- The initial state has 10 possible actions, and each state at each successive level has one less action
 - Average Bits per state: 0.51
 - Flexibility: 33.6%

$$\frac{\sum_{k=0}^{n-1} \frac{n! \log_2(n-k)}{(n-k)!}}{\sum_{k=0}^{n-1} \frac{n!}{(n-k)!}}$$

Idealized Space: Order Matters

- There are ten blocks numbered 1 through 10, which must be dropped into a deep chute in numerical order.
- Here, there is only one possible path.
- There are ten non-terminal states with 1 action per state:
 - Average Bits per state: $10 * \log_2(1) / 10 = 0$ bits
 - Flexibility: 0%

Central Line Flexibility

Space	Average Bits/state	Flexibility
Idealized	0.1	9.1%
Natural	0.94	48.5%
System	0.91	47.6%

Properties of Flexibility Measure

- Captures intuitive notion of flexibility
- Independent of the size of the space
 - Allows comparison of spaces regardless of number of states and paths
- Can work with spaces that have cycles (e.g., undo and reset)
- Natural extension to spaces where the probability of each action is known

Conclusion and Future Work

- Problem space flexibility framework is a promising approach to analyzing and designing SYF systems
- Future work
 - Expand and Refine Framework
 - Empirically evaluate framework by examining low vs. high flexibility tasks in the context of low vs. high flexibility system designs