Optimizing Healthcare Applications Using Systematic-Yet-Flexible System Analysis

Dissertation Proposal

Eliz Markowitz, MS\textsuperscript{1,2}

Committee:

Jiajie Zhang, PhD\textsuperscript{1,2} – Chair
Elmer V. Bernstam, MD\textsuperscript{1,2}
Jorge R. Herskovic, MD, PhD\textsuperscript{2,3}
Todd R. Johnson, PhD\textsuperscript{2,4}

\textsuperscript{1}School of Biomedical Informatics
The University of Texas Health Science Center
Houston, Texas

\textsuperscript{2}National Center for Cognitive Informatics and Decision Making in Healthcare

\textsuperscript{3}The University of Texas MD Anderson Cancer Center
Houston, Texas

\textsuperscript{4}Division of Biomedical Informatics—Department of Biostatistics
The University of Kentucky
Lexington, Kentucky
Optimizing Healthcare Applications Using Systematic-Yet-Flexible System Analysis

Mirroring successes in other industries, efforts to improve health care quality have led to an increased push to develop and adopt systems that enforce or encourage consistent procedures based on best practices and evidence-based medicine. Although such systems can lead to more efficient and safer care, health care is filled with complexity, variations, and exceptions that are not easily captured by idealized processes. Information systems that are too rigid to support such deviations can lead to decreases in quality, along with caregiver resistance and creative workarounds, that together lower the adoption rate and decrease the positive effects of technology [1-5].

Similar problems in other industries have led to the concept of Systematic Yet Flexible (SYF) systems, in which the system supports and encourages a systematic approach, while simultaneously allowing for considerable flexibility [6]. Building upon the general design goals for SYF systems, we designed a framework, called SYFSA (Systematic Yet Flexible System Analysis) for analyzing and designing SYF systems by considering the trade-off between systematicity and flexibility. SYFSA is based on analyzing a task using three related problem spaces: the idealized space, which represents the best practice, the natural space, which captures the possible path actions that may be taken in the real world, and the system space, which specifies how the task is to be done in the system being (re)designed. According to this framework, the most effective system is one that maximizes the chance of idealized performance, as specified by the task space, while also allowing flexible behavior to cope with the inevitable variations in healthcare tasks.

We can use SYFSA to predict performance by comparing the systematicity and flexibility of the task and the system. The long-term goal of this work is to guide the design of systems that allow graceful degradation from ideal performance on standard cases to more flexible performance for coping with variations. Our unifying hypothesis is that SYFSA will allow us to predict user performance. We will test this hypothesis with the following specific aims:

Specific Aim 1: Evaluate the flexibility-compatibility hypothesis. We hypothesize that user performance will improve when the flexibility of the system interface matches the flexibility of the task. We will vary task flexibility and interface flexibility in a 2 × 2 (task flexibility × system flexibility) design and measure user error rates, task completion rates, and task completion times.

Specific Aim 2: Evaluate the prediction that soft constraints that suggest a systematic approach to a task, can provide equal or better overall performance than hard constraints. We hypothesize that soft constraints in a system’s user interface will provide better overall performance than hard constraints, because the soft constraints can support common best practices while still allowing for variations in task demands. Specifically, SYFSA predicts that soft constraints should reflect the most likely course of action for a task.

Specific Aim 3: Evaluate SYFSA predictions that systems that offer less support for the constraints in the idealized task space will lead to longer novice task times, more errors, and more trials to reach expert performance.

Our theoretical contribution is that SYFSA will provide a framework for systematically exploring the flexibility-systematicity trade-offs in information systems. If these predictions hold true, our practical contribution is that using SYFSA for interface design will help improve the quality, safety, and efficiency of health information technology.
RESEARCH STRATEGY

Background and Significance

The fields of Human Computer Interaction and Interface Design are relatively new in relation to many scientific fields. Indeed, much of the early work dedicated to the aforementioned fields has occurred within the past fifty years. While there have been significant and noteworthy developments in these nascent fields, there is still a considerable amount of work to be done.

Contemporary History

Early Human Computer Interaction studies recognized the importance of identifying the goals, objects, relations, constraints, and operations of a task, but did not propose specific methodologies for achieving such objectives. Recent studies have discovered the benefit of defining a Work Domain Ontology (WDO) and an interface with an ideal measure of flexibility [7]. A WDO defines the explicit, abstract, implementation-independent description of the task by separating the task from work context, application technology, and the human cognitive architecture [7]. In other words, the WDO separates the hard constraints and demands of the task from those imposed by the particular tools and workflow used to do the task. In addition to the necessary functionality, it is also important to set the scope of the model by specifying the assumptions. For a WDO, some of the most important assumptions relate to whether or not an artifact is critical or optional in the design space.

Once defined, analysts use a WDO to determine how best to distribute tasks among humans and tools. This distribution defines the basic functionality of a tool and the required functional interactions between the tool and its human user(s). Since the WDO places functionality as top priority in the design process, it is a key factor in the creation of highly efficient and usable interfaces. However, simply defining the WDO for a given task is not enough. In order to create an efficient and usable interface, it is imperative to match interface and task flexibility.

The concept of system or process flexibility has been explored for at least the past few decades. There is an extensive and still growing literature in a multitude of fields including chemical process engineering [8], manufacturing design [9,10], and business process and workflow automation systems [11-14]. While there remains no single, precise definition of flexibility, one general consensus of this work is that flexibility is a multidimensional concept, where the relevant dimensions of flexibility depend on the kind of process or system being analyzed and the analyst’s goals. Furthermore, there exists a general consensus that flexibility is the ability of a system to tolerate and adjust to variations in operating conditions. Based on design constraints required to achieve flexibility, there are often trade-offs between different dimensions; i.e., increased flexibility may lead to decreased efficiency and vice-versa.

With its emphasis on adapting to and tolerating variation, some researchers have argued that the general definition of flexibility implies that there are some invariants that need to be maintained [15]. In short, in the same way that the wings of plane must flex, but still return to their original positions, a flexible system must also be somewhat resistant to change. Some of the more formal definitions and approaches to measuring flexibility operationalize this concept by defining a range of operation that a system must maintain in the face of variation. Flexibility is then the amount of variation that can be tolerated while maintaining operation in the desired range [8].

Working in the context of exploratory data analysis systems, Perer and Shneiderman proposed seven Systematic Yet Flexible (SYF) design goals for systems that support exploratory data analysis [6]. The design goals are to enable users to:

1. See an overview of the sequential process of actions.
2. Step through actions.
3. Select actions in any order.
4. See completed and remaining actions.
5. Annotate their actions.
6. Share progress with other users.
7. Reapply past paths of exploration on new data.

These design goals provide useful advice for tasks that generally require a single sequence of actions, but they do not provide guidance on assessing task flexibility or the trade-offs among user interfaces that support different amounts of flexibility for the same task.

**The Necessity of Extending SYF**

Past research into the fields of Human-Computer Interaction and Interface Design has led to the development of principles and frameworks supporting the idea that the most effective interfaces have structure and organization that match the task itself [16-18]. Efforts to improve the effectiveness and efficiency of Health Information Technology, such as clinical guidelines, structured documentation, standardized terminologies, decision support systems, checklists, and organizational policies, are often too rigid to support the variable nature of healthcare [1-5]. Task-interface mismatch reduces both user efficiency and interface usability. Consequently, it stands to reason that the systematicity and flexibility of an interface should align with the optimal outcome of the task.

Previous work on SYF provides considerable insight on the nature of flexible systems, but neither provides guidelines on how to measure flexibility nor make quantitative, testable predictions. Since SYF design principles have demonstrated improvements in a number of industries [19], the quantitative nature of SYFSA has the potential to improve Health Information Technology by providing information that will allow designers to match system flexibility to task requirements.
Approach — Preliminary Results

Building upon the general design goals for SYF systems, we designed a framework, called SYFSA (Systematic Yet Flexible System Analysis) for analyzing and designing SYF systems by specifically considering the trade-off between systematicity and flexibility. SYFSA is based on analyzing a task using three related problem spaces: the natural space, the idealized space, and the system space. The idealized space represents the best practice—how the task is to be accomplished under ideal conditions. The natural space captures the task actions and constraints on those actions imposed by the physical world, or the current system. The system space specifies how the task is done in the system being (re)designed, including how it may deviate from the idealized space, and how the system supports or enforces constraints that are not present in the natural space. According to this framework, the most effective SYF system is one that maximizes the chance of idealized performance, as specified by best practice, while also allowing flexible behavior to cope with the inevitable variations in healthcare tasks.

SYFSA can be used to make performance predictions based on differences between the systematicity and flexibility of the task compared to the system. The long-term goal of this work is to support the design of systems that allow graceful degradation from ideal performance on standard cases to more flexible performance for coping with the inevitable variations in healthcare.

Central Line Insertion

In order to begin understanding the required systematicity and flexibility of a given task, we analyzed the task of inserting a central line, or a central venous catheter [20]. In health care, central lines are used to deliver medications and fluids, obtain blood samples, and take measurements, such as central venous pressure. While central line infections are a common side effect, the chance of infection is lowered by both following infection control guidelines during and after insertion.

Our example uses a simplified version of the central line insertion procedure. The following actions are listed in the approximate order necessary to comply with the best practices for infection control:

- Sterilize Site
- Drape Patient
- Put Hat On
- Put Mask On
- Put Gown On
- Wash Hands
- Glove Up
- Insert Central Line
- Apply Sterile Dressing

The Idealized Space

In an ideal situation, a caregiver first prepares the patient by sterilizing the insertion site and then fully draping the patient. Next, the caregiver inserting the central line must put on a mask, hat, and gown. Here, the order of operations does not matter; placing the gown on does not prevent the caregiver from putting on the hat, donning the mask does not prevent the caregiver from putting on a gown, etc. Once the aforementioned are on, the caregiver washes her hands and puts on the sterile gloves. Finally, she inserts the central line and then places a sterile dressing over the insertion site.
After identifying the essential operations necessary for the task of central line insertion, we specified the WDO for the task in order to identify the idealized space. For the idealized central line insertion space, we assume that all necessary supplies are available, there is sufficient time to perform the procedure according to best practices, and that a single caregiver will perform the entire task. We then used the following Boolean state representations that record whether or not an action was accomplished:

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

Our operators are defined using logical preconditions on the state and how they change the state (Table 1). In the idealized space, the initial state is one in which

<table>
<thead>
<tr>
<th>Operator</th>
<th>Precondition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sterilize Site</td>
<td>IsterilizedSite</td>
<td>sterilizedSite -&gt; True</td>
</tr>
<tr>
<td>Drape patient</td>
<td>IdrapePatient &amp; &amp; sterilizedSite</td>
<td>drapePatient -&gt; True</td>
</tr>
<tr>
<td>Put Hat On</td>
<td>IhatOn &amp; &amp; drapePatient</td>
<td>hatOn -&gt; True</td>
</tr>
<tr>
<td>Put Mask On</td>
<td>ImaskOn &amp; &amp; drapePatient</td>
<td>maskOn -&gt; True</td>
</tr>
<tr>
<td>Put Gown On</td>
<td>IgownOn &amp; &amp; hatOn &amp; &amp; maskOn</td>
<td>gownOn -&gt; True</td>
</tr>
<tr>
<td>Wash hands</td>
<td>IWashedHands &amp; &amp; gownOn</td>
<td>washedHands -&gt; True</td>
</tr>
<tr>
<td>Glove up</td>
<td>IglovesOn &amp; &amp; washedHands</td>
<td>glovesOn -&gt; True</td>
</tr>
<tr>
<td>Apply Sterile Dressing</td>
<td>IsterileDressing &amp; &amp; centralLineInserted</td>
<td>sterileDressing -&gt; True</td>
</tr>
<tr>
<td>Insert Central Line</td>
<td>IcentralLineInserted &amp; &amp; glovesOn</td>
<td>centralLineInserted -&gt; True</td>
</tr>
</tbody>
</table>

Table 1: Operators and preconditions for the idealized central line insertion space.

Figure 1: The idealized problem space for central line insertion.
nothing has been accomplished (all variables are false) and the goal state is one in which everything has been accomplished (all variables are true). However, this specification of the goal state only guarantees that each action has been taken, not that each action has occurred in the correct order. Since our chosen state representation does not allow for sequence specifications, we constrain the sequence through operator preconditions.

Visualizing the problem space allows one to pinpoint sources of flexibility. Figure 1 depicts the idealized problem space for central line insertion, in which there is a single goal state that can be reached by two different paths. Accordingly, there is very little flexibility in the idealized space.

The Natural Space

Unlike the idealized space, the natural space considers all possible paths and task actions leading to the goal state in the real world. For example, one natural constraint is the inability to remove a surgical glove that has not yet been put on. The natural space also allows us to separate primary and ancillary goals. Here, inserting the central line is the primary goal and applying a sterile dressing is secondary. Furthermore, since the natural space aims to model real world performance, but contains all the actions necessary to mirror the ideal process, it is not necessary to construct a new WDO for the natural space.

Our assumptions for the natural central line insertion space are identical to those of the idealized space, with one exception; here, we define central line insertion as the primary goal and application of the sterile dressing as possible, but not required. Furthermore, the operators are identical to those of the idealized space, but the preconditions reflect hard constraints found in the task environment (Table 2). In this scenario, the hat and mask cannot be put on after the gown and the sterile dressing cannot be put over the insertion site prior to central line insertion. Additionally, the preconditions also reflect the assumption that no other actions, except the application of the sterile dressing, will be taken once the central line is in place.

In the natural space, the initial state is the same as the idealized space and the goal state is any state in which the central line is in place. However, by analyzing the natural space via a network diagram (Figure 2), it becomes apparent that the natural space is more complex and has more apparent flexibility than the idealized space.

As in the idealized space, the initial state is green, goal states are red, and the state with all actions accomplished, regardless of order, is depicted in blue. Since the natural space recognizes that a person may stop once the primary goal state has been accomplished, there are many more goal states in the natural space.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Precondition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sterilize Site</td>
<td>SterilizedSite &amp; IcentralLineInserted</td>
<td>sterilizedSite -&gt; True</td>
</tr>
<tr>
<td>Drape patient</td>
<td>DrapedPatient &amp; IcentralLineInserted</td>
<td>drapePatient -&gt; True</td>
</tr>
<tr>
<td>Put Hat On</td>
<td>haton &amp; IgownOn &amp; IcentralLineInserted</td>
<td>haton -&gt; True</td>
</tr>
<tr>
<td>Put Mask On</td>
<td>lmaskOn &amp; IgownOn &amp; IcentralLineInserted</td>
<td>maskOn -&gt; True</td>
</tr>
<tr>
<td>Put Gown On</td>
<td>IgownOn &amp; IcentralLineInserted</td>
<td>gownOn -&gt; True</td>
</tr>
<tr>
<td>Wash hands</td>
<td>washedHands &amp; IcentralLineInserted</td>
<td>washedHands -&gt; True</td>
</tr>
<tr>
<td>Glove up</td>
<td>IglovesOn &amp; IcentralLineInserted</td>
<td>glovesOn -&gt; True</td>
</tr>
<tr>
<td>Apply Sterile Dressing</td>
<td>IsterileDressing &amp; IcentralLineInserted</td>
<td>sterileDressing -&gt; True</td>
</tr>
<tr>
<td>Insert Central Line</td>
<td>IcentralLineInserted</td>
<td>centralLineInserted -&gt; True</td>
</tr>
</tbody>
</table>

Table 2: Operators and preconditions for the natural central line insertion space.
Indeed, the natural space consists of 384 states, of which 286 are goal states. Furthermore, there are 13,004 paths to any state in which the central line is inserted with the shortest path having one step and the longest having nine steps. Additionally, while there are 1,680 possible paths to the “ideal” state, only two of these paths contain the correct sequence of nine actions that reflect the best practice.

**The Natural Space vs. The Idealized Space**

By utilizing information theory [21] to compare the amount of information required in the natural space versus the idealized space, we can determine how efficiently the natural space supports the best practice. In information theory, the amount of information, measured in bits, between $n$ equally likely actions, is $\log_2(n)$. In short, the total information required to perform a sequence of actions is the sum of the information for each decision along the path.

In the idealized space, there are two correct paths of nine non-terminal states; eight of these states allow a single action and the remaining state has two possible actions. Accordingly, in the idealized state, only one bit of information is necessary to correctly perform the task.

Conversely, the lack of constraints in the natural space means that, while the two ideal paths exist within the natural space, seven of the non-terminal states allow for more than one action. For example, in the natural state, the user can move from the initial state to any of the remaining eight possible actions. From the second state, the user can move to any of the remaining seven actions, et cetera, with each action eliminating one possible state until the last two non-terminal states each permit a single action (having zero bits of information). Therefore, the total bits in either correct path is represented as:

$$\sum_{i=2}^{8} \log_2(i) = 15.2992$$

(Equation 1)

Since the idealized space requires only 1 bit of information, and the natural space requires over 15 bits of information, the efficiency of the natural space for enforcing the best practice is

$$\frac{1}{15.2992} \times 100 \approx 6.5\%$$.

Accordingly, an individual working in the natural space must convey far more information than one in the idealized space in order to achieve optimal performance. Furthermore,
the Hick-Hyman law states that the time needed to make a decision is proportional to the amount of information in the available choices [22, 23]. In conjunction, such information theoretic analysis of a space allows one to both measure cognitive load and estimate relative task times. The characteristics of the natural space, therefore, lead to increased flexibility and a decreased likelihood of achieving idealized task performance. Considering both the idealized and natural spaces allows us to consider a SYF system that encourages ideal performance, while supporting graceful degradation under unexpected or unusual conditions.

**The System Space**

Our system space is based on the intervention proposed and implemented by Berenholtz, et al., which has lowered the incidence of central line related bloodstream infections to nearly zero in numerous institutions [20,24]. The intervention proposed by Berenholtz consisted of five complementary interventions:

1. Educating staff on best practices and the intervention
2. Creating a central line insertion cart to ensure easy access to necessary supplies
3. Asking daily whether or not the central line could be removed
4. A checklist to ensure adherence to best practices
5. Empowering nurses to halt the procedure if guidelines were not followed in non-emergent situations

For this study, we were solely concerned with intervention elements that directly affect the placement of the central line. The central line insertion supply cart supports our idealized space assumption that all necessary supplies are available. Likewise, both the checklist and nurse empowerment encourages, and even enforces, the idealized practice. Such interventions provide the systematicity necessary to adhere to best practices. However, in atypical or emergency situations, the system provides the necessary flexibility to allow the clinician to deviate from best practices.

The system space, therefore, is a combination of both the natural and idealized spaces with an additional root state that switches between the two depending on the nature of the situation. By switching to the natural space, constraints imposed by the idealized space are relaxed, and the provider may choose a non-optimal goal state.

**Information Theoretic Measure of Flexibility**

An information theoretic flexibility measure should capture our intuition regarding flexibility (i.e., systems that we understand to be flexible should have high flexibility) and should allow us to compare flexibility across problem spaces. It is necessary to distinguish between problem space flexibility and inherent flexibility; the former refers to the amount of flexibility in a specific problem space and the latter refers to the amount of flexibility inherent in a task. We define inherent flexibility as the problem space flexibility of the idealized space. Due to the fact that a particular system or natural space may permit incorrect or irrelevant tasks, problem space flexibility may differ from task flexibility. When a system is more flexible than the task, it allows actions that may lead to errors and inefficiencies. Conversely, it may be impossible to complete a task in a system that is less flexible than the task’s idealized space. However, a system design may support more or less flexibility than is inherent in the task since the amount of inherent flexibility in a task may not be required to complete a task. For example, when asked to remember ten random words, it may be more helpful to remember a single sequence of words, even if the sequence of the words is unimportant.

Instead of using the number of paths to the goal to define flexibility, we can use the average bits needed to choose an action per non-terminal state. If there exist $n$ non-terminal states, $S_1, \ldots, S_n$, and
these states have a corresponding number of actions given by $a_1, \ldots, a_n$, the average bits per non-terminal state $F$ is given by:

$$F = \frac{\sum_{i=1}^{n} \log_2 (a_i)}{n} \quad \text{(Equation 2)}.$$  

We can then convert $F$ to percentage flexibility using

$$\frac{100F}{F+1} \quad \text{(Equation 3)}$$

Using these measures, Table 3 shows the flexibility of the three spaces for central line insertion. As expected, the idealized space has the least flexibility, whereas the natural and system spaces have considerably more, with the system space being almost as flexible as the natural space.

<table>
<thead>
<tr>
<th>Space</th>
<th>F</th>
<th>%F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idealized</td>
<td>0.1</td>
<td>9.1%</td>
</tr>
<tr>
<td>Natural</td>
<td>0.94</td>
<td>48.5%</td>
</tr>
<tr>
<td>System</td>
<td>0.91</td>
<td>47.6%</td>
</tr>
</tbody>
</table>

Table 3. Flexibility of three catheter insertion spaces using bits per state (Equations 2 and 3)

However, there is a problem with the latter portion of this analysis. The general problem is that Equation 2 assumes that all states have an equal chance of being visited. This concept is clearly false since actions from any given state may be selected with different probability. If a non-terminal state $S_i$ has $a_i$ actions, and those actions have probabilities $p_1^i, \ldots, p_{a_i}^i$ then a choice of action at $S_i$ conveys an average number of bits given by:

$$B_i = \sum_{j=1}^{a_i} p_j^i \log_2 \left( \frac{1}{p_j^i} \right) \quad \text{(Equation 4)}$$

This results in a version of Equation 2 that considers the probability of actions:

$$F = \frac{\sum_{i=1}^{n} B_i}{n} \quad \text{(Equation 5)}$$

However, this equation still fails to address how action probabilities affect the likelihood of reaching future states. To account for the probabilistic effects of actions on future states, the average bits per state need to be weighted by the probability of reaching each state. If there are $n$ non-terminal states, and these states have probabilities $s_1, \ldots, s_n$, then the weighted average bits per non-terminal state is given by:

$$F = \frac{\sum_{i=1}^{n} s_i B_i}{\sum_{i=1}^{n} s_i} \quad \text{(Equation 6)}$$
Because the probabilities of the non-terminal states do not sum to one, the weights are normalized by dividing by their sum. Table 4 compares the flexibility of the three central line insertion spaces using the non-weighted, non-probabilistic $F$ from Equation 2 to that of Equation 6.

<table>
<thead>
<tr>
<th>Space</th>
<th>Non-Probabilistic (Equation 2)</th>
<th>Weighted Probabilistic (Equation 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F$</td>
<td>%F</td>
</tr>
<tr>
<td>Idealized</td>
<td>0.1</td>
<td>9.1%</td>
</tr>
<tr>
<td>Natural</td>
<td>0.94</td>
<td>48.5%</td>
</tr>
<tr>
<td>System</td>
<td>0.91</td>
<td>47.6%</td>
</tr>
</tbody>
</table>

Table 4. Flexibility of three catheter insertion spaces using non-probabilistic and probabilistic flexibility measures.
Clinical decision support systems, in conjunction with computerized physician order entry tools, have been shown to decrease medical errors, improve efficiency, and enhance clinical performance [25-29]. However, CPOE systems can also cause a number of unintended adverse consequences. [30-33]. We hypothesize that many of the unintended consequences are due to a mismatch between system and task flexibility.

While there are a number of standalone and embedded Computerized Physician Order Entry (CPOE) tools available, the process for medication ordering varies widely among the systems and physicians themselves. However, the process for prescribing medication generally involves selecting the medication, dose, form, frequency, quantity, and number of refills. Furthermore, there are certain limitations to consider. For example, the probability of a clinician assigning either the quantity or refills before selecting a drug is minimal. Accordingly, based on the input of clinicians at The University of Texas Health Science Center, we model the idealized space for medication ordering as relatively linear (Figure 3) having a maximum flexibility of 2 bits; one bit for decision that can occur. Using a version of the SYFSA, which the sum of the bits per path is averaged, the idealized space for medication ordering has an average of 3.5849 bits for the task, 0.5122 average
We analyzed OpenVista (Figure 4), developed by the United States Department of Veteran Affairs and one of the most prevalent EHR systems currently in use, for the medication entry system for high interface flexibility [34]. Figure 6, which depicts the possible paths to the ideal goal state, shows the natural space of OpenVista’s medication entry interface. Using the flexibility measures described in the preliminary studies, we found that the OpenVista problem space has 5040 possible paths [Figure 6] from the initial state to the ideal goal state has an average of 10.6907 bits for the medication ordering task. Furthermore, OpenVista has 1.9313 average bits per decision per path and 65.886% flexibility.

We then analyzed Droogle with RxTerms (Figure 5), developed by Peter Killoran at The University of Texas Health Science Center, for the medication entry system for low interface flexibility. Droogle incorporates a drug library with standardized approximately 24,000 commonly used drugs and, as seen in Figure 7, allows the user only one point of flexibility; the user can decide whether to enter the medication dose prior to the medication form, or vice-versa. Figure 7, which depicts the possible paths to the ideal goal state, shows the natural space of Droogle’s medication entry interface. Using the flexibility measures described in the preliminary studies, we found that the Droogle natural space has 2 possible paths [Figure 7] from the initial state to the ideal goal state having an average of 1 bit per path. Furthermore, Droogle has 0.125 average bits per decision per path and 11.11% flexibility.

![Figure 6: OpenVista – Natural Space](image1)

![Figure 7: Droogle – Natural Space](image2)
PROPOSED STUDIES

Specific Aim 1: Evaluate the flexibility-compatibility hypothesis. We hypothesize that user performance will improve when the flexibility of the system interface matches the flexibility of the task.

Rationale
Based on our preliminary studies, we hypothesize that the ideal flexibility for a system can be identified by matching system flexibility to task flexibility. According to our flexibility-compatibility hypothesis, in an optimal system, the flexibility will match the flexibility of the task. We believe that creating a system with more flexibility than necessary to achieve the optimal goal will lead to inefficiencies and greater errors. Conversely, creating a highly inflexible system may lengthen completion times and cause user fatigue and dissatisfaction.

By determining how flexibility impacts both a given task and interface, we are be able to understand how these dimensions relate to one another. Evaluating the flexibility-compatibility hypothesis will allow us to measure the effectiveness and efficiency of user interfaces and, therefore, guide interface design.

Methods
While the flexibility of medication ordering in OpenVista and Droogle was compared in the preliminary studies, the interfaces differ in more than just flexibility. Indeed, OpenVista and Droogle are vastly different in terms of layout, design, and methodology. Thus, it is necessary that identical medication ordering interfaces are created that differ only in the flexibility allowed to complete a given task. Accordingly, the first step will be to design and implement two medication ordering systems; one having high interface flexibility, hereafter referred to as HIF, and one having low interface flexibility, hereafter referred to as LIF. The basic interface will resemble the low-level prototype seen in Figure 8. HIF and LIF will be built in Java to run as a standalone interface across all platforms. In order to ensure that we maintain the integrity of user error rates and task completion times, algorithms to capture the aforementioned measurements will be built into the code itself.

To test the flexibility-compatibility hypothesis, we will vary task flexibility and interface flexibility in a 2 × 2 design (Table 5), testing the following scenarios:

1. High task flexibility & high interface flexibility
2. High task flexibility & low interface flexibility
3. Low task flexibility & high interface flexibility
4. Low task flexibility & low interface flexibility

We selected the Meaningful Use Cases for CPOE, as stated by the Office of the National Coordinator, to represent low flexibility tasks. These use cases were selected because the information provided reflects the idealized space for medication ordering (Figure 3). As such, the medication order for low task flexibility will be one of the following:

- TD170.304.a – 1.1
  - Amoxil 250 mg oral suspension, Disp 150 ml, Sig: Take 5 ml q8h X 10 days, 0 Refills
  - Plavix 75 mg tablet, Disp #30, Sig: Take 1 tablet QD, 1 Refill
  - Catapres 0.1 mg tablet, Disp #60, Sig: Take 1 tablet bid, 1 Refill

- TD170.304.a – 1.4:

Table 5: 2x2 for Task vs. Interface Flexibility
Glyburide (Diabeta) 2.5 mg tablet, Disp #60, Sig: Take 1 tablet PO, Q AM, 0 Refills
Atorvastatin calcium (Lipitor) 10 mg tablet, Disp #60, Sig: Take 1 tablet PO QD, 1 Refill
Candesartan cilexetil (Atacand), 16 mg tablet, Disp #60, Sig: Take 1 tablet PO QD, 1 Refill

For high task flexibility, we selected an infusion order of Precedex (dexmetetomidine) 0.2 mcg/kg/hr IV in a 72 kg man. There is much more flexibility in this task because the clinician may have to make more decisions regarding the dilution of concentrated medications, the loading dose of the medication, or the units themselves. In this situation, the clinician must switch between screens in the interfaces, allowing much more flexibility in how the task is completed.

We will conduct a controlled trial, in which clinicians will be asked to complete the four scenarios with the appropriate interfaces. We will then measure user error rates, task completion rates, task completion times, and subjective user satisfaction across the four distinct scenarios. Based on a power analysis (Appendix A), in which there is a 99% confidence level and the power is set to 0.97, we would need 13 individuals per sample, or 26 individuals for the entire 2x2 experiment.

Expected Results
According to this framework, the most effective SYF system is one that maximizes the chance of idealized performance, as specified by the task space, while also allowing flexible behavior to cope with the inevitable variations in healthcare tasks. Therefore, we anticipate the following:

1. HIF will perform best in the first scenario, in which both the task & interface are highly flexible.
2. LIF will perform best in the fourth scenario, in which both the task & interface are highly inflexible.
3. Both interfaces will perform adequately in the second and third scenarios.

Figure 8: Low-Level Prototype of Medication Order Interface
Specific Aim 2: Evaluate the prediction that soft constraints, which suggest a systematic approach to a task, can provide equal or better overall performance than hard constraints. SYFSA predicts that soft constraints in a system’s user interface can provide better overall performance than hard constraints, because the soft constraints can support common best practices while still allowing for variations in task demands.

Rationale

As previously shown, medication ordering systems have the ability to impose both hard and soft constraints on the attending clinician. However, when hard constraints are overused, often lead to clinician alert fatigue, and can actually hinder efficiency, or time taken to complete a task, and effectiveness, the ability to perform the task according to best practices. Consider the nurse who intended to program an infusion of nitroglycerin for 5 mcg/minute, but accidently selected a rate of 5 mcg/kg/minute (equivalent to 350 mcg/minute for a 70 kg patient). An alert appeared and the nurse bypassed the warning, a soft constraint [34].

Indeed, many medical technologies utilize soft constraints more often than necessary, which leads to greater chance of alert fatigue; a situation in which users simply click through screens in order to complete a task. Conversely, an interface that works via a sequence of hard constraints may limit the user’s ability to enter certain information, by removing the ability to enter certain information, or force the user to enter information that is not applicable to a given patient.

Methods

The selected tasks and interfaces will be the same as those described in Specific Aim 1. To evaluate the effectiveness and efficiency of interface constraints, we will vary task systematicity and constraint type in a 2 × 2 design (Table 6), testing the following scenarios:

1. High task flexibility & soft constraints
2. High task flexibility & hard constraints
3. Low task flexibility & soft constraints
4. Low task flexibility & hard constraints

Table 6: 2x2 for Task vs. Constraint Type

In this scenario, HIF is the interface with soft constraints. While HIF places the medication ordering fields in the ideal order, there is nothing from stopping a user from skipping back and forth through fields. For example, a clinician in HIF could enter the medication name and then enter the dose, frequency, form, etc. in any order. LIF, however, will only let the user proceed if he/she enters the medication order according to the linear sequence shown in Figure 6. Accordingly, LIF is the interface with hard constraints.

We will conduct a controlled trial, in which a set of clinicians will be asked to complete the selected tasks on two separate user interfaces; one interface uses hard constraints and the second interface uses soft constraints. We will measure user error rates, task completion rates, task completion times, and subjective user satisfaction. We will ask the clinicians to complete the IBM Computer Usability Satisfaction Questionnaire [Figure 9] which gauges user interface satisfaction on a scale from 1 to 7. Through the objective calculations and subjective reviews, we will be able to determine both potential benefits and limitations of utilizing soft and hard constraints in interface design. Based on a power analysis (Appendix B), in which there is a 99% confidence level and the power is set to 1.0, we would need 5 individuals per sample, or 10 individuals for the entire 2x2 experiment.
**Expected Results**
SYFSA predicts that soft constraints in a system’s user interface can provide better overall performance than hard constraints, because the soft constraints can support common best practices while still allowing for variations in task demands. Therefore, we anticipate the following:

1. Average user performance, gauged by task completion time and user errors, is better when completing a task using an interface with soft constraints.
2. User satisfaction will be greater for interfaces with soft constraints than.

**Specific Aim 3:** Evaluate SYFSA predictions that systems that offer less support for the constraints in the idealized task space will lead to longer novice task times and more trials to reach expert performance.

**Rationale**
Based on SYFSA and the Hick-Hyman law, it stands to reason that more time will be required to become an expert user on a highly flexible interface, than it would to become an expert user on an inflexible interface. For example, when entering form dosage of a medication in a CPOE-CDS, constraints can serve as provide user ‘training.’ If such constraints were absent, novices would have no direction, and would have greater error rates and task times.

It also stands to reason that interfaces having an overabundance of flexibility can lead to user dissatisfaction. Accordingly, when an interface provides appropriate guidance, by means of either hard or soft constraints, users will more satisfied than when an interface allows for infinite flexibility.

**Methods**
The selected tasks and interfaces will be the same as those described in Specific Aim 1. We will then perform a Keystroke-Level Model (KLM) analysis on both HIF and LIF to estimate expert user completion time. The purpose of KLM is to determine the functionality of the system and provide estimates of task performance time [35].

**Figure 9:** IBM Computer Usability Satisfaction Questionnaire
Proposed by Card and Moran, the Keystroke-Level Model (KLM) is an offshoot of a GOMS analysis in which the overall execution time of a task is estimated by assigning a time to each operation. The KLM model has six operators [36]:

- K for pressing a key or mouse button
- P for pointing to a location on screen with the mouse
- H for moving hands to home position on the keyboard or to the mouse
- B for a button press
- M for mentally preparing to perform an action
- T(n) for typing a string of characters (n * K seconds)

While the exact time associated with each action in the KLM model varies, Kieras has produced estimated execution times for each action [37]:

- K is 0.20 seconds
- P is 1.10 seconds
- H is 0.40 seconds
- B is 0.10 seconds
- M is 1.20 seconds
- T(n) is 0.20 *n seconds

We will conduct a controlled trial, in which a set of clinicians, unfamiliar with the user interfaces, will be asked to complete a task with two interfaces: one that has fewer constraints than ideal and one with the ideal number of constraints. We will measure user error rates and task completion times. Furthermore, we will record the number of trials necessary for a novice user to achieve expert performance, as determined by the KLM analysis. Based on a power analysis (Appendix C), in which there is a 99% confidence level and the power is set to 0.82, we would need 7 individuals per sample, or 14 individuals for the entire 2x2 experiment.

**Expected Results**
We believe that systems that offer less support for the constraints in the idealized task space will lead to longer novice task times, more errors, and more trials to reach expert performance. Therefore, we anticipate the following:

1. Average user performance, measured by task time and error frequency, declines when using an interface having fewer constraints than deemed appropriate by the idealized space.
2. The rate of learnability will be greater for interfaces that offer less support for the constraints than determined by the idealized space.
3. User satisfaction will be greater for interfaces in which the constraints of the idealized space match the constraints of the interface.
POTENTIAL PITFALLS & ASSOCIATED WORKAROUNDS

Perhaps the biggest potential problem involves ensuring that the interfaces are controlled in all aspects except for flexibility. As noted in the proposed work, the integrity of the experiment would be compromised if the interfaces varied in more than just flexibility. Accordingly, it is necessary that we construct identical interfaces that vary solely in flexibility. One potential issue with constructing the interfaces involves coding the interfaces themselves. While creating prototypes of the interfaces should be relatively straightforward, programming the interfaces to be fully functional may be difficult. Therefore, in the event that the programming aspect of the experiment be outsourced, it is necessary that the storyboarding of the interface be quite detailed and the functionality of each interface is clearly laid out.

Another potential problem may arise in the area of subject recruitment. In particular, the first part of the research, in which task completion time and user error rates are analyzed, requires 26 individuals to attain the desired confidence and precision. There are a number of ways to work around this potential problem:

1. **Reducing the power of the experiment to be less than 0.80.** While the power of the experiment could be lowered in order to reduce the required number of subjects, reducing the power will compromise the integrity of the analysis.
2. **Alter the experiment from a between-groups to a within-groups analysis.** If it becomes clear that the experiment be converted to a within-group analysis, we will need to incorporate unique use-cases in order to control for user learnability. For example, the amount of time it takes an individual to complete a task lowers over time if the user repetitively does an identical task. Thus, rather than having the subjects do the same task for each set, we will need to introduce unique tasks for each individual test. Doing so will reduce the necessary number of subjects to 13 and still provide insight into the relationship between task and interface flexibility.
3. **Altering the target population.** Ideally, all the clinicians could be recruited from The University of Texas-School of Biomedical Informatics. However, recruiting 26 clinicians from a single source may prove problematic. Accordingly, one alternative would be to reach out to other institutions and recruit clinicians from other institutions in the Texas Medical Center. Furthermore, if recruiting clinicians becomes problematic, we could extend the scope of our recruitment to both medical students and nurses.
<table>
<thead>
<tr>
<th>TIMETABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IRB submission</strong></td>
</tr>
<tr>
<td>Acquire infusion pumps</td>
</tr>
<tr>
<td>Task Selection - SYFSA</td>
</tr>
<tr>
<td>Data collection</td>
</tr>
<tr>
<td>Data analysis</td>
</tr>
</tbody>
</table>

**SA1: Evaluate the flexibility-compatibility hypothesis**

**SA2: Evaluate the prediction that soft constraints, which suggest a systematic approach to a task, can provide equal or better overall performance than hard constraints.**

| Task selection | 6/12 | 7/12 | 8/12 | 9/12 | 10/12 | 11/12 | 12/12 | 1/13 | 1/14 |
| SYSFA task analysis | | | | | | | | X | |
| Data collection | | | | | | | | X | |
| Data analysis | | | | | | | | | X |

**Arc Paper (1/3): The Flexibility-Compatibility Principle**

**Arc Paper (2/3): Soft vs. Hard Constraints in interface design.**

**SA3: Evaluate SYFSA predictions that systems that offer less support for the constraints in the idealized task space will lead to longer novice task times and more trials to reach expert performance.**

| Conduct trials | 6/12 | 7/12 | 8/12 | 9/12 | 10/12 | 11/12 | 12/12 | 1/13 | 1/14 |
| Data Collection | | | | | | | | X | |
| Data analysis | | | | | | | | | X |

**Arc Paper (3/3): Less is More—When constraints adversely effect learnability**

| | 6/12 | 7/12 | 8/12 | 9/12 | 10/12 | 11/12 | 12/12 | 1/13 | 1/14 |
| | X | | | | | | | | |
REFERENCES


[38] Card SK, Moran TP, & Newell A. The keystroke-level model for user performance time with interactive systems. Communications of the ACM. 1980; 23(7), 396-410.

APPENDIX A – POWER ANALYSIS FOR SPECIFIC AIM 1

1. Please name your experiment: Task Flex vs. Interface Flex

2. Are you measuring one sample or comparing two?  
   - Measuring One  
   - Comparing Two

   What are the relative sizes of the samples?  
   (You'll set total number of observations later)  
   - A: 50%  
   - B: 50%

3. What is the name of the variable you are measuring?  
   User Errors

   Does the variable have many possible values (height, temperature)? Or just two (positive/negative)?  
   - Many Values  
   - Two Values

   What is the mean value of your variable in Sample A?  
   3.50

   What is the mean value of your variable in Sample B?  
   5.00

   What is the standard deviation of your variable? (Assume same for both samples)  
   1.00

4. Do Samples A and B have different means?  
   Test: Unpaired two sample two tail t-test.

   Current:  
   \[ p = 0.00 \]  
   (100% Cont)  

   Desired:  
   \[ p < 0.05 \]  
   (95% Cont)  

   Set p and power

   Power = 0.97  
   > 0.80

   Difference between A and B:  
   1.50 ± .776

   Number of observations:  
   28  
   13

   Observations needed for desired confidence and precision

   Samples significantly different

---

Estimates of Averages of Samples

- Sample A  
  - User Errors  
  - [t Distribution](#)

- Sample B

Estimate of Difference between Sample Means

- Shaded area is 95% confidence interval
APPENDIX B – POWER ANALYSIS FOR SPECIFIC AIM 2

1

Please name your experiment: Task Flexibility vs. Constraint Type

2

Are you measuring one sample or comparing two?

Measuring One  Comparing Two

What are the relative sizes of the samples? (You’ll set total number of observations later)
A: 50%  B: 50%

3

What is the name of the variable you are measuring?
User Satisfaction

Does the variable have many possible values (height, temperature)? Or just two (positive/negative)?

Many Values  Two Values

What is the mean value of your variable in Sample A? 2.20
What is the mean value of your variable in Sample B? 3.60

What is the standard deviation of your variable? (Assume same for both samples)
1.60

4

Current:

\[ p = 0.00 \]  (100% Conf)  \[ < \]  0.05  (95% Conf)

Desired:

\[ > \] 0.50

Power = 1.00

Do Samples A and B have different means?
Test: Unpaired two sample two tail t-test.

Confidence Interval (95%) of Difference Between Samples
[1.204, 1.996]

Set p and power

Difference between A and B: 1.60  ± .396

Number of observations: 100 5

Observations needed for desired confidence and precision

Samples significantly different

5

Estimates of Averages of Samples

Sample A  Sample B  t Distribution

Estimate of Difference between Sample Means

Shaded area is 95% confidence interval
APPENDIX C – POWER ANALYSIS FOR SPECIFIC AIM 3

1. Please name your experiment: Learnability

2. Are you measuring one sample or comparing two?
   - Measuring One
   - Comparing Two

   What are the relative sizes of the samples? (You'll set total number of observations later)
   - A: 50%
   - B: 50%

3. What is the name of the variable you are measuring?
   - Task Completion Time

   Does the variable have many possible values (height, temperature)? Or just two (positive/negative)?
   - Many Values
   - Two Values

   What is the mean value of your variable in Sample A?
   - 17.20

   What is the mean value of your variable in Sample B?
   - 18.00

   What is the standard deviation of your variable? (Assume same for both samples)
   - 0.40

4. Do Samples A and B have different means?
   - Current: \( p = 0.01 \) (99% Conf)
   - Desired: \( p < 0.05 \) (95% Conf)
   - Power = 0.82

   Confidence Interval (95%) of Difference Between Samples:
   - \([0.24, 1.358]\)

   Difference between A and B:
   - 0.80

   ± .558

   Number of observations:
   - Sample A: 11
   - Sample B: 7

   Observations needed for desired confidence and precision

   Samples significantly different

---

Estimates of Averages of Samples
- Sample A
- Sample B

Estimate of Difference between Sample Means
- Task Completion Time
- Shaded area is 95% confidence interval