The usability canary in the security coal mine: A cognitive framework for evaluation and design of usable authentication solutions

Brian Glass*, Graeme Jenkinson†, Yuqi Liu*, Angela Sasse*, and Frank Stajano†
*University College London
†University of Cambridge

Abstract—**Over the past 15 years, researchers have identified an increasing number of security mechanisms that are so unusable that the intended users either circumvent them or give up on a service rather than suffer the security. With hindsight, the reasons can be identified easily enough: either the security task itself is too cumbersome and/or time-consuming, or it creates high friction with the users' primary task. The aim of the research presented here is to equip designers who select and implement security mechanisms with a method for identifying the "best fit" security mechanism at the design stage. Since many usability problems have been identified with authentication, we focus on "best fit" authentication, and present a framework that allows security designers not only to model the workload associated with a particular authentication method, but more importantly to model it in the context of the user's primary task. We draw on results from cognitive psychology to create a method that allows a designer to understand the impact of a particular authentication method on user productivity and satisfaction. In a validation study using a physical mockup of an airline check-in kiosk, we demonstrate that the model can predict user performance and satisfaction. Furthermore, design experts as well as novices suggested personalized order recommendations which were similar to our model's predictions. Our model is the first that supports identification of a holistic fit between the task of user authentication and the context in which it is performed. When applied to new systems, we believe it will help designers understand the usability impact of their security choices and thus develop solutions that maximize both.**

1. Introduction

Over the past 15 years, the security community has started to acknowledge that security mechanisms are only effective if they are usable: users frustrated by overzealous security measures bypass the security if they can, or switch to a competing system that is easier to use. While an increased awareness of the damage that lack of usability can inflict is a first step, in practice security experts and developers who choose security mechanisms have no way of gauging what the impact of their choice on users will be—and most are not able to call on a human usability expert to do this for them. There are tools for developers to carry out walkthroughs and assessments of a particular solution. The time it will take a user to complete a task can be estimated using the Keystroke Level Modelling (KLM-GOMS) model [1], and an automated version CogTools [2] provides such a prediction from screen interaction with the tool. This approach, however, has limitations:

1) It only supports evaluation and comparison of specified solutions, rather than discovery of the "best" one, and
2) it does not take account of the impact that different mental and physical tasks have on subsequent tasks.

In this paper, we contribute and validate an intellectual tool—a design and evaluation framework—that will help designers gain a better understanding of the cost of security, with specific reference to user authentication. Our framework and methodology assesses security mechanisms not in isolation but in the context of the so-called *primary task* that constitutes the user's true goal. What users really want (primary task) is to check in for a flight or pay a bill, not recall and enter a password or read off and transcribe a one-time code. From the users' perspective, these are distractions imposed in the name of security, often to manage threats they don't know exist.

The cost of a given security measure, such as entering a password, is not absolute: it is instead also a function of its relationship to the other components of the primary task. A recent study [3] found that authentication creates a "wall of disruption" in users' work. This is not only the time spent on the security task, but the knock-on effect of re-starting the primary task after an interruption. Thus, the cost depends not just on how hard the authentication task is in itself but also on when it occurs in the users workflow, on what functions of the brain it loads and on what else the user was meant to be doing before and after.

We draw on results from cognitive psychology to assess the cost of task switching between different activities. Our framework lets designers model the tasks of the intended scenario and the precedence constraints that describe their relationships, and then quantitatively compares alternatives to suggest combinations that minimize the cognitive load and usability cost to the user.

In addition to providing this novel methodology, we present a validation study which verifies the tool's insights. Using a physical mockup, we test the tool's optimal ("best") suggestion against its pessimal (”worst”) suggestion. Moreover, we surveyed a group of professional designers to test our tool's automatic suggestions against the intution of human experts.

2. Modelling a business process

A business process (or workflow) is a collection of interrelated tasks that are performed by users in order to achieve some objective. It is often the case that only authorized users may perform certain tasks: in such cases the business process will include one or more tasks requiring explicit user authentication. Tasks that require authentication impose ordering constraints on the business process (users shouldn't be able to complete a task requiring authorization until they have been authenticated). More generally, the business process may have some freedom in the order in which tasks are performed, that is, the tasks have a partial order. In such cases, system designers have flexibility to rearrange tasks to maximise the system's usability.

Our goal in modelling a business process is twofold. Firstly, we wish to determine the optimal ordering of the tasks, taking into account the switching costs described in sections 4.2 and 4.3. Secondly, we wish to explore the impact of equivalent but alternative tasks for user authentication. Thus, our model of a business process must include:

- A representation of the set of steps to be performed,
- A set of tasks that can be performed at each step,
- Hard constraints that enforce the partial ordering of the tasks, and
- Soft constraints that capture the costs of switching between tasks.

2.1. Example: airport check-in kiosk

Throughout this paper our example will be airport check-in using a self-service kiosk. We are not modelling the kiosk of any particular airline or airport but an imaginary one that combines features we have observed on a variety of real kiosks. We use this business process as our example because its tasks, listed in Table 1, use a range of different cognitive resources, detailed in Table 4. We include cognitive tasks such as making decisions or selections and carrying out checks, as well as physical tasks like attaching luggage tags. The check-in procedure necessarily also includes some form of authentication, but there are multiple ways of achieving that. Helping a designer select the most appropriate authentication mechanism for a specific business process is one of the goals of our framework.

We are also interested in finding the optimal order for the tasks. The check-in kiosk example exhibits a reasonable degree of ordering flexibility. Figure 1 shows the dependencies between the check-in tasks.

3. Our framework

The framework we present allows developers to assess the usability of different security tasks within a workflow

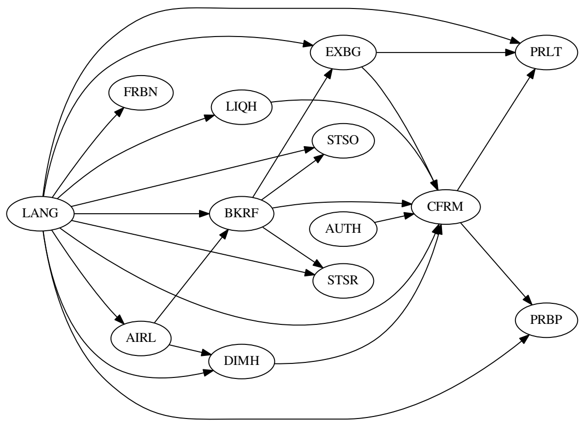

Figure 1. Dependencies between Airport self-service check-in tasks. An edge from node u to node v indicates that u must be carried out before v. For example, users must enter their booking reference (BKRF) sometime before they confirm their check-in (CFRM).

such as the check-in example described above. Two overarching principles inspired this framework:

1) Assessing the usability of an individual task is important, but insufficient, and
2) the order in which tasks appear can have an interactive, global effect on overall usability.

Specifically, these principles imply that swapping out one authentication method for another may have carryover effects on the overall workflow. They also suggest that an automated optimization procedure could be used to “solve” for the optimal ordering of tasks—the one that minimises cognitive interruptions and maximizes usability.

A workflow has some number of “steps” and the user can carry out exactly one task at each step. We want to find the optimal assignment of tasks to steps, respecting any ordering constraints between the tasks—ensuring, for example, that certain tasks happen after the authentication task. We will present a method for encoding a workflow as a weighted constraint satisfaction problem (WCSP) [4], in which there are a set of variables (the steps), a set of values (the tasks) and a set of constraints. Further information about constraint satisfaction problems is given in section 5 and the means of encoding a workflow is explained in section 5.2. Workflow environments that are designed to accomplish a specific goal (*e.g.*, withdrawing cash from an ATM, or checking in at an airport kiosk) can be conceptualized as a sequence of tasks completed in a linear fashion. Transitioning from one task to another will carry an additional transition cost. The total usability of an overall task is thus a combination of the costs of the individual tasks and the costs of the pairwise transitions between the tasks in the linearized sequence. Task ordering can have a potentially unpredictable impact on the entire task workflow. Considering the usability of many different potential orderings is a non-trivial task but a

Task	Code	Prerequisites	Description
Select language	LANG		User selects their preferred language from the displayed options.
Select airline	AIRL	LANG	User selects their airline from the displayed options.
Booking reference	BKRF	LANG, AIRL	User enters their booking reference using an touchscreen QWERTY keyboard.
Authenticate	*AUTH*		User authenticates their identity.
▷ Passport scan	AUPS	LANG	User authenticates by scanning the photo page of their passport.
▷ Passport information	AUPI	LANG	User authenticates by manually entering their passport information.
▷ Insert payment card	AUCC	LANG	User authenticates by inserting their payment card.
▷ Password	AUPW	LANG	User authenticates by typing a password (assuming the user has an account with the airline).
Check forbidden items	FRBN	LANG	User presses a button to confirm that their luggage doesn't contain any of the displayed items.
Check liquids	LIQH	LANG	User presses a button to confirm that their hand luggage doesn't contain any containers of liquid above a certain volume.
Check luggage size	DIMH	LANG, AIRL	User presses a button to confirm that their hand luggage is below a certain size.
Select outbound seat	STSO	LANG, BKRF	User selects their outbound seat by clicking on a plan of the available seats in the airplane.
Select return seat	STSR	LANG, BKRF	User selects their return seat by clicking on a plan of the available seats in the airplane.
Buy extra bag	EXBG	LANG, BKRF	User optionally pays for additional luggage by clicking a button and swiping a credit card.
Confirm	CFRM	LANG, BKRF, *AUTH*, LIQH, DIMH, EXBG	User confirms the details entered so far by reading some text and pressing a button.
Print luggage tag	PRLT	LANG, EXBG, CFRM	User takes a luggage tag from the machine and attaches it to their luggage.
Print boarding pass	PRBP	LANG, CFRM	User takes a boarding pass from the machine.

TABLE 1. AIRPORT SELF-SERVICE CHECK-IN TASKS.

computerized tool that computes optimal ordering solutions makes it tractable.

The concept that reordering tasks can have an effect on usability comes from established principles in cognitive psychology. An established literature exists on the relative ordering effects of different types of tasks [5]. In section 4 we explain how these effects were operationalized from available literature. In this section, we give an overview of how to extend existing assessments of workload by considering not only the endogenous task demands but also the additional exogenous demands that emerge from the transitional costs between tasks.

3.1. Completion times

Various methods have been devised to predict the time required to complete a given task. A popular technique is KLM-GOMS [1]. In this technique, the designer breaks down the task into a variety of individual action components (for example: mentally prepare, click button, press a key), each of which has an associated reaction time. This technique is useful for estimating how long it would take a user to complete a given task. Another assessment technique is CogTool [2], which assesses task completion times and learning rates based on shifting visual attention and making motor responses. Both methods use approximations for mental processes (*think* in CogTool, *mental preparation* in KLM-GOMS). In the present paper, we seek to expand on these techniques by assessing the differential cognitive demands of different tasks as well as task transitions.

3.2. Cognitive demands of tasks

While subjective measures of workload are useful tools in predicting user satisfaction and adoption rates, the operationalization of workload as a unitary resource does not fit with modern theories of cognition [6], [7]. Rather, a variety of dissociable mechanisms underlie cognition and become active given characteristics of the task at hand [8]. In section 4.2.2 we address the various cognitive mechanisms involved in an individual task.

3.3. Cognitive demands of transitions

While tasks carry their own demands, there are also certain performance costs associated with switching from one task to another. These transitional costs can be asymmetric; that is, switching from Task A to Task B may be more costly than switching from Task B to Task A [5]. For this reason, we have coded principles of task switching costs from existing literature. In section 4.2 and section 4.3 we address the various types of switch costs used in the present modelling procedure.

3.4. Quantifying tasks

The goal of our work is to promote a discipline for considering both the unary and transitional demands of tasks on users, and to demonstrate a method for improving performance by minimizing overall task demand. The effectiveness of any given instantiation of this methodology depends directly on the quality of input information about the workflow being analysed. Thus, it will be crucial to develop a valid and reliable regimen for quantifying task characteristics. In this initial paper we are charting a new path and, for illustrative purposes, we have assigned numerical values based on our judgement. In future work we would develop instruments such as worksheets and flowcharts to help independent designers assign consistent and reproducible numerical values when they assess their tasks.

4. Cognitive psychology

4.1. Task switching

When a person switches from one task to another task, the brain must reorganize and reallocate cognitive resources to ensure an efficient transition [5]. Transitioning from a task that primarily uses resource A to a task that primarily uses resource B (instead of continuing to use resource A) results in performance deficits, or switch costs. Experimental psychology has uncovered certain principles that govern these transitions. These so-called *switch cost asymmetries* have been shown to occur, or not, depending on other characteristics of the tasks involved. We have codified these task asymmetries (expressed as Cohens d effect sizes, which are a commonly used metric in psychology for comparing the mean of one sample to that of another [9]) into a collection of rules that may be encoded as constraints in a weighted constraint satisfaction problem (see section 5). Below, we describe how we constructed these rules from available literature on switch cost asymmetries. The rules fall into two categories: *cognitive resource transitions* and *task property transitions*.

4.2. Cognitive resource transitions

One reason that task switching results in a performance deficit is the requirement for the individual to disengage active cognitive mechanisms and then engage other cognitive mechanisms in order to match task demands [5]. For example, switching from a visual task to an auditory task is more costly than vice versa [10]. In a practical example, if a person were performing a hypothetical two-factor authentication procedure that involved recognizing an image among several on a large screen and also recognizing a voice over a phone line, it could be more efficient to place the audio identification subtask before the visual authentication subtask. This demonstrates that task ordering can impact user efficiency due to asymmetries in cognitive switch costs.

4.2.1. Cognitive resources demands of individual subtasks. The cognitive mechanisms included in the present implementation are visual working memory (VWM; responsible for holding, processing, and operating on information of immediate importance), procedural memory (PM; responsible for storing and preparing motor action sequences), declarative recall (DR; responsible for generating and presenting stored information on demand), semantic recognition (SR; responsible for determining whether factual information has been stored in memory), and episodic recognition (ER, responsible for determining whether information about experienced events have been stored in memory). Note that while the categories represented here have an empirical basis, the taxonomy of mental processes is a fluid research topic [11].

Table 2 reports the costs of switching between tasks utilising different cognitive mechanisms. The values are Cohens d effect sizes and were calculated from published studies [9] involving empirical measurements of reaction time in various task switch contexts, which assessed the efficiency with which individuals were able to transition between different cognitive systems.

4.2.2. Operationalizing the check-in task. In order to utilise these principles of task switch cost asymmetry, we operationalised identified the cognitive resources most likely to be engaged by the subtasks involved in the Airline Self-Service Machine Task. While this is a first approximation, in the future empirical methods could be used to verify these predictions. In real-world tasks, many different cognitive mechanisms are likely to be engaged simultaneously. For our purposes, we have selected the dominant resource which is predicted to have the highest relative engagement level. Table 4 reports the major cognitive resource assigned to each subtask, as well as the physical response modality, voluntary/involuntary nature, task familiarity, and task complexity.

It is impractical to determine the specific brain networks activated for a specific real task, so we characterize each task by assessing its similarity to documented cognitive tasks. For example, determining whether a piece of hand luggage exceeds certain dimensions is similar to documented tasks involving assessing geometric attributes of three dimensional shapes, a task known to activate visual working memory [12]. This is a tractable simplification of the reality of cognitive functioning for two reasons:

1) Real-world tasks likely engage many different cognitive mechanisms at once, with varying degrees of demand. For our purposes we consider the cognitive mechanisms deemed to be most relied upon in order to complete the task.
2) Many other cognitive mechanisms exist than were included in Table 2. For simplicity, we only included the primary mechanisms involved for each task. Future implementations could include other systems such as auditory working memory.

4.3. Task property transitions

An important source of task switch costs is the impact of the interference or inertia carried over from one to another. One counterintuitive finding is that switching from a less familiar task to a more familiar task is actually more disruptive than vice versa [13]. The prevailing reasoning behind this effect is that when engaged in a less familiar task, the individual must suppress commonly used mental processes in lieu of less frequently used processes [14]. This suppression has a carry-over effect on the new task, resulting

		To				
		VWM	PWM	DR	SR	ER
From	Visual working memory (VWM)	0	0.495	0.495	0.495	0.157
	Procedural memory (PM)	0.495	0	0.495	0.699	0.699
	Declarative recall (DR)	0.495	0.495	0	0.482	0.482
	Semantic recognition (SR)	0.495	0.842	1.078	0	0.433
	Episodic recognition (ER)	0.307	0.842	1.078	0.354	0

TABLE 2. COSTS OF SWITCHING BETWEEN TASKS UTILISING DIFFERENT COGNITIVE MECHANISMS, GIVEN AS COHENS d EFFECT SIZES.

in a performance deficit. These transitional asymmetries have also been identified when transitioning between tasks that differ by complexity [15], recent practice [13], modality (form or method of response) [16], and whether the task was voluntary [17]. These empirical observations have been codified into conditional rules with associated effect sizes in Table 3.

4.3.1. Complexity. Task complexity was assessed using existing definitions from experimental psychology [15], namely the number and combination of rules required to solve or complete the task. For example, subtraction is relatively less complex than division. The reason for this is that division uses the principles of subtraction as well as other principles, such as remainders and carrying digits between places. In the airline check-in task, for example, the task regarding forbidden materials was considered to be more complex than the task regarding liquids. This is because it is more complex to determine whether several items fall into several categories versus a single category.

4.3.2. Familiarity. Task familiarity was determined by assessing not only the frequency with which an average user completes a given task, but also whether the task assesses familiar knowledge or processes [13]. For example, selecting your language preference might not necessarily be a common chore, but it requires judgment based on a familiar fact. In contrast, printing a luggage tag is something that is an activity that is both infrequent and unfamiliar.

4.3.3. Response Modality. Response modality refers to the physical method for issuing a response from the user to the system. For example, different modalities include a QWERTY keyboard, a mouse pointer, or a verbal response. There is evidence that transitioning from one response modality to another can incur a switch cost. However, Sandhu and Dyson [16] demonstrate that a switch cost due to response modality may not occur when a modality switch coincides with a cognitive resource switch. In other words, switching response modalities is most disruptive when it is the only change that takes place.

5. Modelling a business process as a constraint satisfaction problem

5.1. Constraint satisfaction problems

The goal of a constraint satisfaction problem (CSP) is to assign *values* to a set of *variables* subject to a set of *constraints*. The constraints express *local* restrictions, such as "these two variables must have different values", but there exist algorithms [18] for finding *global* solutions. Below we shall describe *weighted* constraint satisfaction problems: these include "soft" constraints that may be violated for some cost. We first introduce the classic CSP framework.

5.1.1. Classic CSP. A *classic* CSP is defined by a triple $P = (X, D, C)$. X is the set of variables, $\{x_1, ..., x_n\}$. A domain $d_i \in D$ is a set of allowable values for variable x_i. A constraint $c \in C$ is a pair (X_c, R_c), where $X_c \subset X$ is the *scope* of the constraint and R_c is a relation over the corresponding set of domains. R_c specifies tuples of simultaneously-allowed values for the variables in the scope and can be defined explicitly as a subset of the product of the domains, or as an abstract relation which can test whether a given tuple of values is allowed, for example: $x_1 \neq x_2$.

An *assignment* specifies values for some or all of the variables. An assignment is *consistent* if it does not violate any constraints. A *complete* assignment is one which assigns values to all variables. A *solution* to a CSP is a complete consistent assignment. A CSP is consistent if a solution for it exists. Finding a solution to a CSP is an NP-complete problem.

5.1.2. Weighted CSP. In a classical CSP the constraints are all absolute or "hard", no consistent assignment can violate any constraint and all solutions are equally "good". Several variants have been proposed to extend the CSP framework to include "soft" constraints expressing priorities, preferences, costs, and probabilities. Schiex, Fargier and Verfaillie [4] generalised these and defined *valued CSP* (VCSP). A VCSP is similar to a classical CSP except that the constraints assign *costs* to assignments instead of allowing or disallowing them[1].

A VCSP is defined by a tuple $P = (S, X, D, C)$, where X and D are sets of variables and their domains as previously. Costs are specified using a *valuation structure*, which is a triple $S = (E, \oplus, \succ)$, where E is a set of costs ordered by $\succ$ and $\oplus$ is an associative commutative monotonic binary operation on E for combining costs.[2] Weighted CSP (WCSP) is a specific subclass of valued CSP in which the costs are the natural numbers or positive infinity, $E = \mathbb{N} \cup \{\infty\}$ and $\oplus$ is the standard sum operation.

1. Equivalently a VCSP can be seen as classic CSP in which each constraint has been annotated with a cost for removing it [4].

2. A classical CSP can be expressed as a VCSP with $E = \{t, f\}$, $\bot = t \succ f = \top$ and $\oplus = \wedge$.

Rule name	Condition	Cost (effect size)
Modality	A switch occurred which uses the same resources (on-diagonal above) **and** there is a modality switch.	0.16
Recent Practice	A task of similar modality or resource has been used anywhere previously.	0.31
Familiarity	The current task is more familiar than the previous task.	0.42
Complexity/Choice	A task is done voluntarily **and** the complexity decreases.	2.92
	A task is involuntary **and** the complexity decreases.	1.63

TABLE 3. Additional costs of transitioning between tasks determined by specific rules, given as Cohens d effect sizes.

Code	Primary cognitive resource	Modality	Voluntary?	Familiarity	Complexity
LANG	Semantic recognition	Touchscreen	No	5	1
AIRL	Episodic recognition	Touchscreen	No	5	1
BKRF	Visual working memory	Touchscreen QWERTY	No	3	3
AUPS	Procedural memory	Passport scanner	No	2	2
AUPI	Procedural memory	Touchscreen QWERTY	No	2	3
AUCC	Procedural memory	Credit card reader	No	3	2
AUPW	Declarative recall	Touchscreen QWERTY	No	4	3
FRBN	Semantic recognition	Touchscreen	No	2	3
LIQH	Episodic	Touchscreen	No	3	3
DIMH	Visual working memory	Touchscreen	No	2	4
STSO	Visual working memory	Touchscreen	Yes	2	4
STSR	Visual working memory	Touchscreen	Yes	2	4
EXBG	Episodic	Touchscreen	Yes	2	2
CFRM	Episodic	Touchscreen	No	4	2
PRLT	Procedural memory	Luggage tag	No	1	5
PRBP	Episodic	Touchscreen	Yes	4	2

TABLE 4. Properties of the check-in kiosk tasks. Familiarity and complexity are on a scale from 1 (low) to 5 (high).

In this framework, constraints specify local costs of assignments. A constraint $c \in C$ is a pair (X_c, F_c) where X_c is its scope and F_c is a cost function,

$$F_c : \prod_{x_i \in X_c} d_i \to E \tag{1}$$

Note that a hard CSP constraint $c = (X_c, R_c)$ can be represented in a WCSP as $c' = (X_c, F_{c'})$, where

$$F_{c'}(v) = \begin{cases} 0 & \text{if } v \in R_c \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

Given a WCSP $P = (S, X, D, C)$, an assignment A of variables $Y \subset X$ has total cost $V_P(A) \in E$. This cost is the sum of all applicable cost functions.

Given a WCSP, the typical task is to find the optimal solution, the complete assignment with the minimum total cost. The most popular algorithms for solving WCSP employ branch and bound search, although algorithms for solving WSCP remain an active research area.

5.2. Our model

As described in section 5.1.2, a weighted CSP is represented by the tuple $P = (S, X, D, C)$. In our model, a business process with n steps (where 1 is the first step performed by the user and n the last) is represented by a set of variables X, $\{x_1, ..., x_n\}$. The domain D (the set of values that can be assigned to variable x_i) consists of all of the tasks, including any user authentication tasks, in the business process. The set of constraints C includes hard constraints that ensure tasks are performed exactly once and ordering relations between tasks are maintained. C also includes soft constraints represent the costs of switching between tasks.

5.2.1. Implementation of our model. A proof-of-concept implementation of our model has been created in Numberjack, a Python framework for constraint programming, mixed integer programming and satisifiability solvers [19]. Numberjack integrates a number of third-party, open source solvers (which are typically written in C/C++ for efficiency) and can be easily extended to include additional solvers. The Numberjack framework currently support Toulbar2—an exact combinatorial optimization tool designed for solving Weighted Constraint Satisfaction Problems (otherwise known as Cost Function Networks) [20]. Numberjack's proposition of a high-level modelling framework and an underlying efficient and high-pedigree solver[3] make it well suited to our purpose.

As shown below, a Numberjack VarArray is used to represent each step in the business process. The domain of each variable is the natural numbers 0...d where each value represents one of the possible tasks. A constraint is then added to the model to ensure that each value in the domain is assigned to exactly one variable.

```
from Numberjack import VarArray

# Create a variable array,
# one variable for each step
# in the business process
wcspVars = VarArray(0, d, nSteps)

model.add(AllDiff(wcspVariables))
```

3. Toulbar2 was a wining solver in the Uncertainty in Artificial Intelligence (UAI) 2010 Approximate Inference Challenge.

A custom Numberjack constraint has been created to enforce the partial ordering of tasks. This constraint (shown below) ensures that for all combinations of the variables in the CSP it is never the case that the value *after* is assigned to a variable that precedes a variable assigned the value *before*.

```
class Order(Predicate):

    def __init__(self, vars, before, after):
        Predicate.__init__(self, vars,
            "Order")
        self.set_children(vars)
        self.before = before
        self.after = after
        self.lb = None
        self.ub = None

    def decompose(self):
        return [(x != self.after) | (y !=
            self.before) for x, y in
            combinations(self.children, 2)]
```

As defined in section 5.1.2, a constraint $c \in C$ is a pair (X_c, F_c) where X_c is its scope and F_c is a cost function. Task switching costs are modelled as binary constraints; that is, their scope is limited to variables that are immediately next to each other. The task switching costs are represented by a d-by-d matrix (where $d = |D|$).

```
from Numberjack import PostBinary

def pairwise(iterable):
    a, b = tee(iterable)
    next(b, None)
    return izip(a, b)

# d-by-d matrix,
# binaryCost[d1][d2] specifies the
# cost of assigning d1 and d2 to
# variables that are immediately
# next to each other
binaryCosts = [...]

for var, varNext in pairwise(wcspVars):
   model.add(PostBinary(var, varNext,
       binaryCosts))
```

5.2.2. Results of modelling the airline self service check-in. Table 5 shows the optimal task ordering given by the solver for the self-service check-in scenario. The four columns of the table correspond to the four different concrete authentication tasks we are considering. The cost reported for each workflow is the sum of all the task switch costs (Cohen's d effect sizes) for that workflow[4]. The fact that the four orderings and total costs are different supports the central message of this paper: fitting an authentication task to its context is important. Specifically, we can see that the passport scan (AUPS) and insert payment card (AUCC) authentication methods yield substantially lower total switching costs—*regardless* of their intrinsic costs. More generally, with twelve task switches in total, the mean cost for each task switch, in each of the four cases, is approximately 0.5, which constitutes a "medium" effect size under the standard Cohen's d interpretation: this indicates that task switches are not an insignificant cost in general.

It is interesting to note that the solver splits the two seat selection tasks for the outgoing and return flight. Within the model, the two selection tasks are indistinguishable so the cost of switching from either to the other is zero. Therefore, we might expect that the solver would place these task next to each other. However, this is an interesting example of how our intuition can be wrong as this local optimization ultimately precludes the globally optimal solution.

4. To obtain the total cost, we should add to that the costs of the individual subtasks. We cannot do that yet, because they are expressed in different non-comparable units, so this is a topic for future research. See the next section, 5.2.3.

5.2.3. Limitations of our model.

Essentially, all models are wrong, but some are useful. —*George E. P. Box* [21]

The first significant limitation of our model is its inability to relate the reported total task switching costs to an additional amount of *time* required to complete the business process. Whilst this is a significant limitation, we feel that the outputs of the model remain useful and may be used alongside the existing techniques for estimating the time taken to carry out specific tasks such as KLM-GOMS.

Secondly, although the cognitive resource transition costs and task property transition costs are based on empirical results from the literature, user studies should be undertaken to validate the way in which they combine within our framework.

As well as splitting up the two seat selection tasks, in three cases the solver has placed return seat selection before outbound seat selection. While this would obviously be somewhat confusing for users, it is understandable that the solver has arranged the tasks in this way because within the model they appear identical. Our model simply doesn't capture the notion that when tasks relate to events that are ordered, it makes sense for those tasks to have the same order. In such cases the system designer must apply their discretion to ensure that the system remains consistent with reality and with user expectations.

6. Validation Study

In order to test the model's predictions, we completed a validation study. Our intention was to validate the theoretical predictions regarding task switching, and thus we focused on the subtasks which would be inherent in airline check-in kiosks regardless of further authentication mechanisms used (e.g., credit card, passport). Using a mock-up of the airline check-in kiosk described above, we sought to assess the model's optimal subtask ordering recommendation. We accomplished this in four ways: 1) Participants completed the optimal (”best”) ordering in a simulated airline departure scenario, 2) These same participants offered their own order

	Select language	Select language	Select language	Select language
	Select airline	Select airline	Select airline	Select airline
	Check liquids	Check liquids	Check liquids	Check liquids
	Booking reference	Booking reference	Booking reference	Booking reference
	Check forbidden items	**Insert payment card**	**Passport info**	**Password**
	Select return seat	Buy extra bag	Select return seat	Check forbidden items
	Check luggage size	Select return seat	Check luggage size	Select outbound seat
	Passport scan	Check luggage size	Check forbidden items	Check luggage size
	Buy extra bag	Check forbidden items	Buy extra bag	Buy extra bag
	Confirm	Confirm	Confirm	Confirm
	Print boarding pass	Print boarding pass	Print boarding pass	Print boarding pass
	Select outbound seat	Select outbound seat	Select outbound seat	Select return seat
	Print luggage tag	Print luggage tag	Print luggage tag	Print luggage tag
Cost	5.53	5.88	8.18	8.42

TABLE 5. Optimal task ordering of the self-service check-in using different authentication mechanisms.

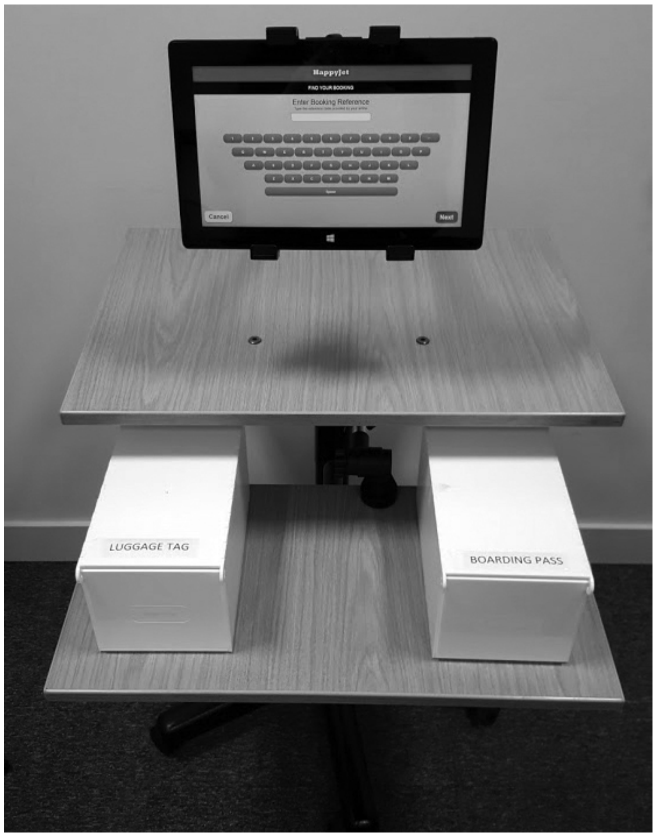

Figure 2. Mock up for the self-service airport check-in kiosk.

recommendations for the task, 3) We further tested a second sample of participants with the pessimal (”worst”) ordering, and 4) We surveyed professionals trained in design fields in order to gather an expert based ordering recommendation. The Optimal ordering was: AIRL, LIQH, BKRF, FRBN, STSO, DIMH, EXBG, CFRM, PRBP, PRLT; the Pessimal ordering was: FRBN, AIRL, BKRF, EXBG, LIQH, DIMH, CFRM, PRLT, STSO, PRBP.

6.1. Participants

Participants were recruited from the University College London student and staff community and compensated £7 for their time. The study was approved by the UCL Ethics Committee, and all participants offered informed consent. For the Optimal condition, 40 participants were recruited. A sample of 20 participants was recruited for the comparative Pessimal condition, and a further 50 self-reported design professionals were recruited to generate the Expert ordering suggestion. The demographics of the group were as follows: Optimal group (Age_{Mean} = 26.6, Age_{SD} = 7.2, 28 females), Pessimal group (Age_{Mean} = 29.1, Age_{SD} = 13.5, 15 females), Expert designers (Age_{Mean} = 30.0, Age_{SD} = 9.7, 12 females, 8 no gender specified). Two participants were removed from the Optimal group for not completing the task, and three were removed from the Expert group for not completing the survey. Participants were asked about their average annual number of flights: $OptimalGroup =$ 4.7($SD =$ 3.4), $PessimalGroup =$ 3.5($SD =$ 3.2).

The sample of Expert designers was recruited from NCR Corporation (www.ncr.com) as well as via the online survey system Prolific Academic (www.prolific.ac), and were selected using a pre-screening occupation questionnaire. The group identified as working with user experience design in physical settings (n=17), software/web settings (n=33), or both (n=6), with 5.2 mean years of experience (SD = 6.0). These participants were compensated with £1.67 for completing the task (equivalent to £5/hour).

6.2. Procedure

6.2.1. Check-in Kiosk. Participants were asked to use the simulated airline check-in kiosk as if they were actually preparing for a departure at an airport. Participants were given two suitcases, one large suitcase for checked baggage, and one small suitcase for carry on. The experimenter opened the small suitcase and described the contents to the user: two shirts, two paperback books, and a plastic bag containing toiletries under 100ml in volume. The experimenter told the participant that the large suitcase contained clothes and no hazardous or forbidden materials. The participants completed the airline check-in kiosk three times, each time with a different provided cover story (given in pseudo-random order between participants). The mock airlines were "MetroAir", "HappyJet", and "QuickFly", and the mock destinations were Glasgow, Edinburgh, and Cardiff (departing from London). Participants took the two suitcases and entered a second room to interact with a kiosk comprised of a touchscreen monitor and two flapped dispensers (one for boarding pass, one for baggage tag) on a small roller table (see Figure 2). The flapped dispensers were pre-loaded with the relevant boarding pass and baggage tag, and a simulated printing sound oriented the participant to their locations during the appropriate subtask. After completing each of the three simulated check-in procedures, the participant moved to a different room and completed the subjective satisfaction questionnaire.

6.2.2. Subjective Satisfaction Questionnaire. After each trial, participants completed the following 13-item Satisfaction Questionnaire [22]. Each item was scored using a 5-point Likert scale (from "Strongly disagree" to "Strongly agree"). In order to reduce repetitiveness, the second and third repetitions of the questionnaire asked for changes in assessment relative to the previous trial (from "Less than before" to "More than before"). In this way, a change score was computed using responses from the first trial as a baseline.

1) The system was annoying to use.
2) I liked using the system.
3) The system did what I thought it would do.
4) The system was fun to use.
5) The system was unreliable.
6) I was satisfied using this system.
7) I was comfortable using this system.
8) The system was disappointing.
9) The system was engaging.
10) The system was unpredictable.
11) I feel positive about the system.
12) I would not want to use this system.
13) The system was pleasant to use.

6.2.3. Ordering Preference Task. After the completion of the check-in procedure, participants were asked to generate their own suggested orderings for the subtasks. Using a computerized tool, participants dragged boxes representing the various subtasks into their preferred orderings. First, participants were allowed to freely order the subtasks without partial ordering constraints. Second, participants were told which subtasks violated the partial ordering constraints (if any), and were asked to rearrange the subtasks until the ordering satisfied the constraints (see Figure 6).

6.3. Results

6.3.1. User Performance. Task performance was measured by calculating the time to complete each subtask. The time was computed based on the duration from completion of previous subtask to the completion of the current subtask.

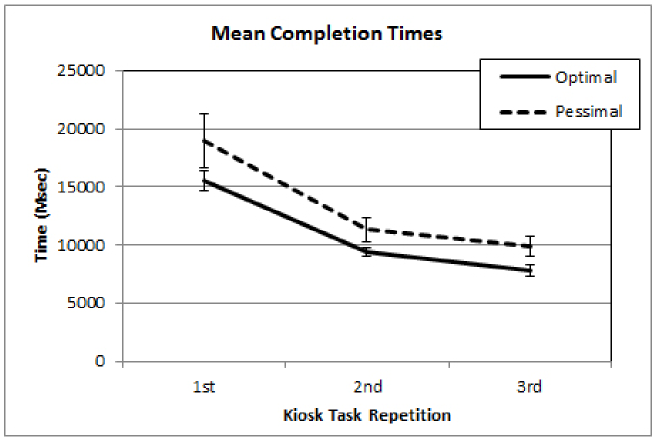

Figure 3. Mean completion times over three task repetitions between Optimal ordering and Pessimal ordering conditions.

Results were similar when time was calculated as the duration from the completion of the previous subtask to the first click of the current subtask, although some subtasks only required one click, thus we present subtask completion times here.

To evaluate the impact of our model's ordering suggestions as well as the impact of prior kiosk experience, participants were further clustered into two experience groups: Have used airline check-in kiosk in the previous calendar year (Used Kiosk), or have not (No Kiosk). Learning curve (repetition over the three trials) was also evaluated as a within subjects factor. Performance (mean completion time) was evaluated using a repeated measures ANOVA with a 2 (Condition: Optimal, Pessimal) x 2 (Experience: Used Kiosk, No Kiosk) x 3 (Repetition) factorial design. There were significant main effects of Condition ($F_{1,55} = 4.82, p = 0.03$) and Experience ($F_{1,55} = 5.01, p = 0.03$) such that those in the Optimal order had faster completion times, and those with airline kiosk experience in the previous year had faster completion times. There was a significant main effect of Repetition ($F_{2,110} = 81.0, p < 0.001$) consistent with a monotonic learning curve (see Figure 3). There was also a significant interaction of Repetition and Experience ($F_{2,110} = 5.09, p = 0.01$) such that those with experience demonstrated a flatter learning curve due to faster initial completion times (see Figure 4). Completion time was lower for 8 out of 10 subtasks (essentially tied for PRBP and AIRL). According to the binomial distribution, the probability of a result at least this extreme occurring from randomly generated data is 5.3%. In summary, those in the Optimal ordering condition demonstrated faster completion times on all three repetitions of the task, and those with prior experience were overall faster as well.

6.3.2. User Satisfaction. User satisfaction was measured using the 13-item Satisfaction Questionnaire (see above) by taking the average responses on a 5-point Likert scale (reverse coded for the negatively worded items). For the second and third completion of the questionnaire, the scores

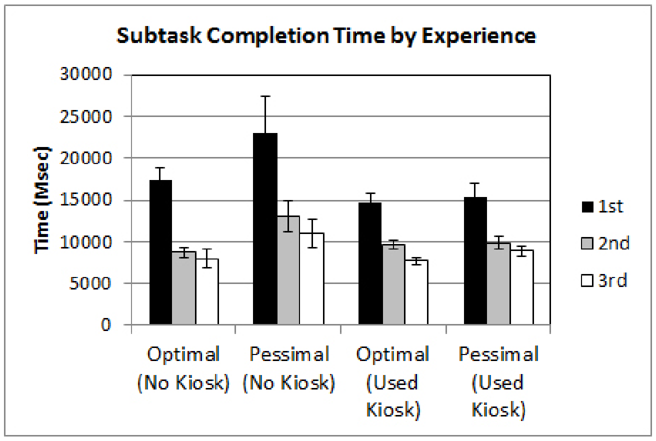

Figure 4. Mean completion times over three task repetitions between Optimal ordering and Pessimal ordering conditions, by experience.

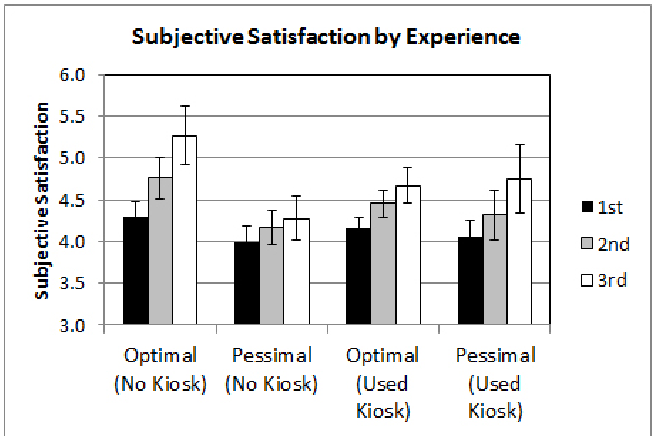

Figure 5. Satisfaction Scores over three task repetitions between Optimal ordering and Pessimal ordering conditions, by experience.

were demeaned (subtracted by 3) and added to the previous questionnaire's result. Satisfaction was evaluated using a repeated measures ANOVA with a 2 (Condition: Optimal, Pessimal) x 2 (Experience: Used Kiosk, No Kiosk) x 3 (Repetition) factorial design. Although directionally in favor of the Optimal ordering, the satisfaction ratings were not statistically significantly higher for the Optimal ordering versus the Pessimal ordering ($F_{1,55} = 2.15, p = 0.149$). There was a significant main effect of Repetition ($F_{2,110} = 27.9, p < 0.001$) such that subjective user satisfaction increased monotonically over the three task repetitions. There was a significant three-way interaction of Repetition, Condition, and Experience ($F_{2,110} = 3.68, p = 0.03$). Figure Figure 5 illustrates the nature of this interaction, such that those with no kiosk experience were more sensitive to the Optimal vs. Pessimal manipulation than those with kiosk experience. Specifically, those with no kiosk usage in the previous year found the Optimal ordering to be more satisfactory over time relative to the Pessimal ordering.

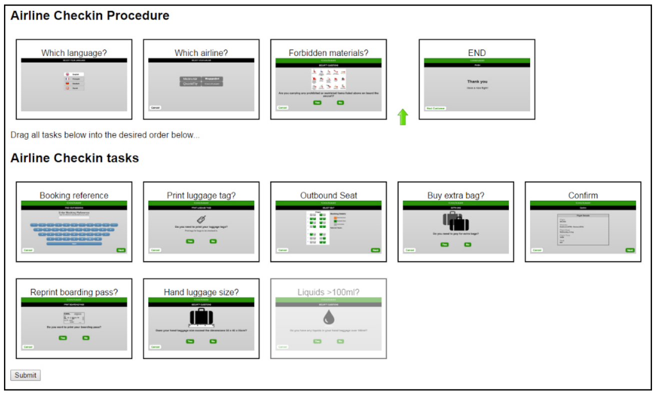

Figure 6. Screenshot of the ordering preference task.

6.3.3. Ordering Preferences. Participants provided their own recommended orderings for the kiosk subtasks. Separately, self-reported design professionals (who did not complete the kiosk task) also provided recommended orderings. From these experts, a consensus Expert ordering was generated (AIRL, BKRF, STSO, DIMH, FRBN, LIQH, EXBG, CFRM, PRLT, PRBP) using the mode frequencies from each subtask index. A Euclidean distance metric (based on index differences) was computed for each participant's recommended ordering. In this way, we were able to calculate a participant's suggestion's difference from the model's Optimal ordering, Pessimal ordering, and an Expert ordering. The Expert ordering was significantly more similar to the model's Optimal ordering than the Pessimal ordering ($t_{Paired} = 9.15, p < 0.001$). Thus, Experts suggested orderings which were more similar to the model's Optimal suggestion.

Prefered ordering was evaluated using a repeated measures ANOVA with a 2 (Condition: Optimal, Pessimal) x 2 (Experience: Used Kiosk, No Kiosk) x 3 (Comparison Source: Optimal, Pessimal, Expert) factorial design. There was a significant main effect of Comparison Source ($F_{2,110} = 27.4, p < 0.001$) and a significant interaction of Comparison Source and Condition ($F_{2,110} = 7.92, p = 0.001$) such that those who participated in the Optimal ordering gave suggestions which were more similar to both our Optimal ordering and the Expert ordering. In summary, the Expert suggested order and the model's Optimal suggested order were closer to recommendations given by participants who had experienced the Optimal ordering (see Figure 7).

7. Conclusion and further work

We presented a framework for reasoning about the impact of user authentication on the overall usability of a workflow. Our framework is the first to highlight the importance of the fit between a particular user authentication method and the context in which it is performed. Specifically, we draw on results from cognitive psychology to quantify the impact of switching between tasks that draw on different cognitive resources and use different modalities.

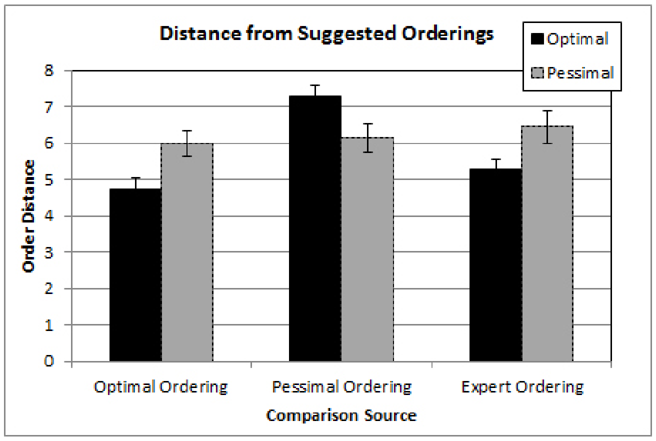

Figure 7. Difference of participants' suggested orderings to the model's Optimal and Pessimal ordering, as well as an Expert suggested ordering.

This is a new, disruptive approach to evaluating usability of security solutions, and even systems usability in general. We are sharing this powerful core idea with the community in its preliminary form but we envisage further work in several directions, both on our proof-of-concept implementation of the solver and on the framework itself. We need to develop reliable input tools, such as worksheets and flowcharts, to allow independent designers to perform consistent assignment of numerical values to the features of their tasks. More fundamentally, we would like to develop a "unit" (not necessarily just elapsed time; maybe other factors like stress and annoyance might come into it) to measure the usability cost, and a disciplined and justifiable method for expressing in this same unit both the cost of a task and the additional cost of a transition. This will allow the CSP solver to add those sub-costs to compute a globally optimal solution. These additional steps go hand in hand with user studies and validation of the modelling approach. But the general principles and methods that underlie our framework are already useful and applicable today.

Our framework targets two audiences: designers of secure systems and designers of new authentication schemes. System designers can use the framework as scaffolding that supports the overall design process. This scaffolding encourages the designer to think about how their use of authentication is likely to impact on their users and ultimately on the success of the system. Similarly, security researchers developing new authentication primitives can use the framework to reason about their solution within a realistic context of use.

Importantly, the theoretical model output was further validated with a user study. Participants performed better in the optimal ordering, and were more satisfied by the optimally ordered interface. The model's optimal ordering was more similar to the suggested orderings of professional designers, and participants who experienced in the optimal ordering were more likely to further prefer and recommend such an ordering. In this way, we were able to validate the predictions of the theoretical model.

The consolidation of results from cognitive psychology on the effects of task switching, and the presentation of these results in a format directly usable by security professionals is perhaps the most useful contribution of our work.

8. Related work

Sasse *et al.* [3], [23] present their findings of a 2-part study into the impact of authentication on the productivity of employees in a US governmental organisation. They conclude that the overall burden of user authentication includes a disruption to the user's primary task (that is, what they are actually trying to achieve). Disruptions resulting from user authentication damage productivity and result in significant frustrations. Furthermore the authors found that *avoidance*—not logging into services or using them less frequently—was an increasingly common coping strategy when the burden of authentication was felt to be too great.

While Shay *et al.* [24] have attempted to boost security by pushing the limits of user workload, there is a call for designers to consider the impacts of effortful authentication mechanisms on the user. Employees reported to Inglesant and Sasse [25] that they'd resort to insecure workarounds in response to increasingly stringent password policies. This friction [26] between the tasks has been shown to moderate individual compliance.

Building on these observations, our work is the first attempt to develop a model of such costs. The ultimate goal of this model is to empower system designers to reason about such effects before deployment.

Prior work has demonstrated the usefulness of modelling subtask arrangement to find optimal orderings. Crampton [27] arranges security-related subtasks to find orderings that satisfy entailment, cardinality, and role-based constraints. Zhang *et al.* [28] use an optimization procedure to minimize mouse clicks in a computerized task workflow. Our methodology uses similar techniques to consider a finer grained user-centric cost model to optimize the handing off of cognitive mechanisms throughout a task.

Constraint Satifaction Problems (CSP) have long found application in decision supports systems. Scheduling—determining the optimum allocation of shared resources to competing activities—is a well-known NP-complete Constraint Satisfaction Problem (CSP) [29].

Cohen *et al.* [30] apply techniques from CSP to the *Workflow Satisfiability Problem (WSP)*—that is, deciding whether a plan exists for assinging task to authorized users in a given business process . Our work draws inspiration from their use of CSP. However, in our framework we are concerned with an optimization problem.

Acknowledgments

The Cambridge authors are grateful to the European Research Council for funding this research through grant StG 307224 (Pico). The UCL authors are grateful to the Engineering and Physical Sciences Research Council for funding this research through grant #EP/K033476/1. We are also grateful to Jeunese Payne and Max Spencer for useful discussions, and to Ben Wong for scheduling and running participants. The authors also thank Graham Johnson for assistance in reaching out to professional designers.

References

[1] B. E. John and D. E. Kieras, "The goms family of analysis techniques: Tools for design and evaluation," Tech. Rep., 1994.

[2] B. E. John, E. W. Patton, W. D. Gray, and D. F. Morrison, "Tools for predicting the duration and variability of skilled performance without skilled performers," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56, no. 1. SAGE Publications, 2012, pp. 985–989.

[3] M. A. Sasse, M. Steves, K. Krol, and D. Chisnell, "The great authentication fatigue–and how to overcome it," in *Cross-Cultural Design*. Springer, 2014, pp. 228–239.

[4] T. Schiex, H. Fargier, G. Verfaillie *et al.*, "Valued constraint satisfaction problems: Hard and easy problems," *IJCAI (1)*, vol. 95, pp. 631–639, 1995.

[5] A. Kiesel, M. Steinhauser, M. Wendt, M. Falkenstein, K. Jost, A. M. Philipp, and I. Koch, "Control and interference in task switchinga review." *Psychological bulletin*, vol. 136, no. 5, p. 849, 2010.

[6] D. de Waard and B. Lewis-Evans, "Self-report scales alone cannot capture mental workload," *Cognition, Technology & Work*, vol. 16, no. 3, pp. 303–305, 2014.

[7] J. De Winter, "Controversy in human factors constructs and the explosive use of the nasa-tlx: a measurement perspective," *Cognition, technology & work*, vol. 16, no. 3, pp. 289–297, 2014.

[8] F. G. Ashby and W. T. Maddox, "Human category learning 2.0," *Annals of the New York Academy of Sciences*, vol. 1224, no. 1, pp. 147–161, 2011.

[9] C. J. Ferguson, "An effect size primer: A guide for clinicians and researchers." *Professional Psychology: Research and Practice*, vol. 40, no. 5, p. 532, 2009.

[10] T. Strobach, R. Liepelt, T. Schubert, and A. Kiesel, "Task switching: effects of practice on switch and mixing costs," *Psychological Research*, vol. 76, no. 1, pp. 74–83, 2012.

[11] A. Baddeley, "Working memory: theories, models, and controversies," *Annual review of psychology*, vol. 63, pp. 1–29, 2012.

[12] P. E. Downing, "Interactions between visual working memory and selective attention," *Psychological Science*, vol. 11, no. 6, pp. 467–473, 2000.

[13] N. Yeung and S. Monsell, "Switching between tasks of unequal familiarity: the role of stimulus-attribute and response-set selection." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 29, no. 2, p. 455, 2003.

[14] M. Gade and I. Koch, "The influence of overlapping response sets on task inhibition," *Memory & Cognition*, vol. 35, no. 4, pp. 603–609, 2007.

[15] J. S. Rubinstein, D. E. Meyer, and J. E. Evans, "Executive control of cognitive processes in task switching." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 27, no. 4, p. 763, 2001.

[16] R. Sandhu and B. J. Dyson, "Modality and task switching interactions using bi-modal and bivalent stimuli," *Brain and cognition*, vol. 82, no. 1, pp. 90–99, 2013.

[17] C. M. Arrington and G. D. Logan, "Voluntary task switching: chasing the elusive homunculus." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 31, no. 4, p. 683, 2005.

[18] V. Kumar, "Algorithms for constraint-satisfaction problems: A survey," *AI magazine*, vol. 13, no. 1, p. 32, 1992.

[19] E. Hebrard, E. O'Mahony, and B. O'Sullivan, "Constraint Programming and Combinatorial Optimisation in Numberjack," in *Proceedings of the 7th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR-10)*, ser. Lecture Notes in Computer Science, A. Lodi, M. Milano, and P. Toth, Eds., vol. 6140. Bologna, Italy: Springer-Verlag, May 2010, pp. 181–185.

[20] D. Allouche, S. de Givry, and T. Schiex, "Toulbar2, an open source exact cost function network solver," Technical report, INRIA, Tech. Rep., 2010.

[21] G. E. Box and N. R. Draper, *Empirical model-building and response surfaces.* John Wiley & Sons, 1987.

[22] A. L. Comrey, "Factor-analytic methods of scale development in personality and clinical psychology." *Journal of consulting and clinical psychology*, vol. 56, no. 5, p. 754, 1988.

[23] M. Steves, D. Chisnell, A. Sasse, K. Krol, M. Theofanos, and H. Wald, "Report: Authentication diary study," 2014.

[24] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor, "Can long passwords be secure and usable?" in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 2014, pp. 2927–2936.

[25] P. G. Inglesant and M. A. Sasse, "The true cost of unusable password policies: password use in the wild," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 383–392.

[26] A. Beautement, M. A. Sasse, and M. Wonham, "The compliance budget: managing security behaviour in organisations," in *Proceedings of the 2008 workshop on New security paradigms*. ACM, 2009, pp. 47–58.

[27] J. Crampton, "A reference monitor for workflow systems with constrained task execution," in *Proceedings of the tenth ACM symposium on Access control models and technologies*. ACM, 2005, pp. 38–47.

[28] Y. Zhang, R. Padman, and J. E. Levin, "Reducing provider cognitive workload in cpoe use: optimizing order sets." *Studies in health technology and informatics*, vol. 192, pp. 734–738, 2012.

[29] D. S. Johnson, "The np-completeness column: An ongoing guide," *Journal of Algorithms*, vol. 3, no. 2, pp. 182–195, 1982.

[30] D. Cohen, J. Crampton, A. Gagarin, G. Gutin, and M. Jones, "Iterative plan construction for the workflow satisfiability problem," *J. Artif. Intell. Res. (JAIR)*, vol. 51, pp. 555–577, 2014. [Online]. Available: http://dx.doi.org/10.1613/jair.4435