Error in Skilled Performance: A Control Model of Prescription Writing

Richard S. Marken

DRU-2751-P1

February 2002

Prepared for Pfizer, Inc.

RAND Health

The RAND unrestricted draft series is intended to transmit preliminary results of RAND research. Unrestricted drafts have not been formally reviewed or edited. The views and conclusions expressed are tentative. A draft should not be cited or quoted without permission of the author, unless the preface grants such permission.
This report describes an approach to evaluating the causes of error in prescription writing. The methodology uses a computer model of prescription writing to determine the likely effectiveness of different prescribing error prevention interventions. The model is designed to exhibit the kind of behavior that is seen in prescription writing and has generalized parameters of broad significance that can be varied to see their effects on error rate. The model described in this report was developed as part of a project aimed at evaluating proposed standards for electronic prescribing (e-prescribing) systems, which have been promoted as one means of reducing prescription errors. The model makes it possible to determine the e-prescribing features that are likely to be most effective in terms of error reduction.
CONTENTS

Preface ................................................................. iii
Figures .............................................................. vii
Tables ................................................................. ix
Summary .............................................................. xi
Acknowledgments .................................................... xiii
1. INTRODUCTION ....................................................... 1
2. A MODEL OF PRESCRIBING ......................................... 1
3. MODEL IMPLEMENTATION ........................................ 3
4. MODEL COMPUTATIONS ........................................... 4
   Closed loop control ........................................... 5
   Skill and Agility .............................................. 6
5. PRESCRIPTION PERCEPTION ...................................... 7
6. WRITING A PRESCRIPTION ........................................ 8
   Prescription completion rule ................................ 9
   Speed and Accuracy .......................................... 10
7. CONTROL ERROR AND PERFORMANCE ERROR .................. 10
   Defining a prescribing error .................................. 11
   Categorical decisions ....................................... 11
8. ERROR RATE AND SKILL LEVEL .................................. 12
9. MODEL VALIDATION .............................................. 14
   Quantitative Validation ..................................... 14
   Qualitative Validation ...................................... 15
10. MODEL EXCURSIONS ............................................... 16
    Time Available to Write a Prescription .................... 16
    Disturbance Magnitude .................................... 16
    Range of Prescriptions ................................... 17
11. TYPES OF PRESCRIBING ERROR ................................ 17
    Slips .......................................................... 17
    Mistakes ...................................................... 18
    Failures ....................................................... 19
12. ERROR PREVENTION ............................................. 19
    Reducing the range of prescriptions ...................... 20
    Changing Work and Workflow ................................ 20
    Electronic prescribing systems ............................ 20
13. DISCUSSION ..................................................... 21
References .......................................................... 23
FIGURES

FIGURE 1: Prescription component writing control system............ 2

FIGURE 2: Spreadsheet organization of prescription-writing model..... 4

FIGURE 3: Behavior of error (S-P), prescription component (q) and component specification (S) over time ................. 8

FIGURE 4: Relationship between control system skill level (gain) and prescription error rate ........................................ 13
TABLES

TABLE 1: Observed rate of different types of prescription error (top row) and rates of the same types of error produced by prescription-writing model. ......................... 14
SUMMARY

A control model of skilled performance is used to explain the rate at which prescribing errors occur. Model error rate depends on skill level, system design characteristics, the range of different types of prescriptions that are written and the time available to complete each prescription. The model produces error rates that are quantitatively equal to those found in studies of the incidence of different types of prescribing error. The model also produces error rates that are qualitatively consistent with the results of studies that show increases in prescribing error rates as a result of increases in distractions and workload. The model was used to determine the likely effectiveness of different prescribing error prevention interventions.
ACKNOWLEDGMENTS

This study was supported by a contract from Pfizer, Inc. to RAND. We would like to thank William T. Powers, Keith Hendy, and Phillip Farrell for comments and suggestions on the model and Shan Cretin for her support and encouragement during the development of the model.
1. INTRODUCTION

Psychologists have developed models that explain the causes of human error (Reason, 1990; Baars, 1992) but these models do not explain why these causes operate only rarely in skilled human performance. Prescribing errors, for example, occur in 0.4 - 1.9% of all medication orders written (Dean et al, 2000; Lesar et al, 1997). Models of human error (Norman and Draper, 1986; Rasmussen, 1986) have explained such errors in terms of system design characteristics but these models do not explain why these design characteristics cause an error only 0.4 - 1.9% of the time. The models tell us the factors that lead to error but they do not tell us how often these factors will actually produce an error. It is, therefore, impossible to use these models to predict error rates or to determine the effect of error reduction interventions on error rate.

This paper describes a control model of human performance that not only explains why errors occur but also why they occur at a particular rate. The model has been used successfully to explain several different kinds of skilled behavior (Marken, 2002). In the present application, the model is used to explain how a skilled physician writes prescriptions. The model is described in terms of plausible brain mechanisms that might underlie prescribing behavior but the model is not intended to be an explanation of how the brain writes prescriptions. Rather, it is a working model that is designed to exhibit the same kind of behavior as that seen in prescription writing, and that has generalized parameters of broad significance that can be varied to see their effects on error rate. The model makes it possible to predict error rates as well as the effect of error reduction interventions on these error rates. The model was originally developed to explain skilled, "errorless" behavior (Powers, 1998) but in the present application it is extended to include the "ability" to make errors at a specified rate.

2. A MODEL OF PRESCRIBING

Writing a prescription is seen as a process of carrying out several intentions at the same time: the intention to write an order for
an appropriate drug at an appropriate dosage via the appropriate route (oral, intravenous, etc.) and so on. The model represents these intentions as separate specifications for each component of a prescription. The model's representation of a physician's intention to write one particular prescription component, the dose of the medication being prescribed, is shown in Figure 1. The horizontal line in the Figure separates the model of the physician from the model of the environment in which the physician acts. Note that variables associated with the physician are capitalized and those that are part of the physician's environment are in lower case.

![Diagram](image)

**Figure 1.** Prescription component writing control system.

In Figure 1, S represents the intention to prescribe a particular drug dose. This intention, which exists in the physician's brain, is represented as a quantitative specification for the dosage that should be written on the prescription. The dose that is actually written on the prescription is represented by the variable q. S is, therefore, a mental specification for the value of the physical variable, q. The physical variable - the dosage being written - is represented in the physician's brain as a perception, P. The physician compares this perception of the dosage that is written to the specification, S, of the dosage that should be written. The dosage that is written -- the value of q -- depends on the physician's outputs, O, which are the thoughts and actions the physician uses to produce the intended dosage. The written dosage also depends on the effects of unpredictable (and often
undetectable) disturbances, d, such as leaky pens, slippery surfaces and similar drug names. Any difference between the physician’s perception of the dosage that is written, P, and what the physician intends to write, S, is an error, E, that causes further outputs, which bring the dosage into alignment with what is intended.

The model in Figure 1 represents writing a prescription component as a dynamic closed loop process. All variables in the model, including the component of the prescription that is being written, are changing over time. The physician acts by producing outputs, O, that bring the prescription component, as perceived, P, to the specified state, S, protecting it from the effects of disturbances. This process is called control and the prescription component being written, q, is called a controlled variable. This control process is driven by error, which is the difference between the intended and perceived state of the controlled variable (E = S - P). Control is a closed loop process in which error drives the outputs that continuously reduce the error that caused those same outputs. The control process is continuous over time. The process stops only when stopped by an outside agency, such as another control system.

3. MODEL IMPLEMENTATION

The prescribing model is implemented using Visual Basic algorithms in an Excel spreadsheet. The layout of the spreadsheet model is shown in Figure 2. The model implements four separate control systems, which produce four different components of a prescription. The control systems are implemented in the columns of the spreadsheet labeled “Drug”, “Dose”, “Route” and “Other”. The “Drug” system carries out the intention of writing the appropriate drug name. The “Dose” system carries out the intention of writing the appropriate dosage for that drug. The “Route” system carries out the intention of writing the appropriate route for the drug (capsule, suppository, etc.). The “Other” system carries out the intention of writing any other necessary information relevant to taking the drug, such as how long the drug should be taken and whether or not the prescription can be refilled.
<table>
<thead>
<tr>
<th></th>
<th>Drug</th>
<th>Dose</th>
<th>Route</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>9.00</td>
<td>21.00</td>
<td>31.00</td>
<td>37.00</td>
</tr>
<tr>
<td>P</td>
<td>9.22</td>
<td>20.89</td>
<td>31.04</td>
<td>37.13</td>
</tr>
<tr>
<td>O</td>
<td>9.05</td>
<td>20.32</td>
<td>30.32</td>
<td>36.64</td>
</tr>
<tr>
<td>q</td>
<td>9.05</td>
<td>20.86</td>
<td>31.02</td>
<td>37.12</td>
</tr>
<tr>
<td>d</td>
<td>0.09</td>
<td>0.48</td>
<td>0.70</td>
<td>0.54</td>
</tr>
<tr>
<td>g</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>s</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0007</td>
</tr>
</tbody>
</table>

**Figure 2. Spreadsheet organization of prescription-writing model.**

The bold horizontal line in Figure 2 represents the boundary between the prescription-writing control systems (which are in the physician's brain) and the physician's environment, where the actual prescription is produced. Each cell in the row labeled S contains a number that corresponds to each system's intention, which is a numerical specification of the prescription component to be produced. Each cell in the row labeled P contains a number that corresponds to each system's perception of the current state of the prescription component being written. Each cell in the row labeled O contains a number that corresponds to the state of the output (action) each system is using to get the prescription component it is controlling into the intended state. The values in the cells of the row labeled q represent the current state of each prescription component. The state of each component at any point in time is assumed to be the combined result of the physician's output and environmental disturbances acting at that time. Disturbances are represented as sinusoidal variations over time. These disturbance values are in the cells of the row labeled d. A different sinusoidal disturbance is applied to each prescription component, q.

### 4. Model Computations

The state of a prescription component at any particular time is represented in the model as the sum of the control system's output, O, and the value of the disturbance, d:

\[ q = O + d \]  

(1)

Note that the "=" sign in equation (1) and all subsequent equations represents the replacement operator rather than mathematical equality.
Equation (1), which is found in each cell of row q, defines the environmental relationships in the model. All the variables in equation (1) are in the model physician's environment. Equation (1) says that the state of a prescription component — what is being written on the prescription pad — is, at any instant, the combined result of the effects produced by the actions of the physician (O) and by environmental factors that are independent of the physician (D).

The values in row S are integer constants that represent the intended state of the perception of the prescription component controlled by each control system. The value of the perception, P, is assumed to be proportional to the value of the prescription component, q, that exists in the physician's environment, so that:

\[ P = k \times q \]  

(2)

The output, O, produced by each control system is defined as:

\[ O = O + s \times (g \times (S - P) - O) \times dt \]  

(3)

where g and s are constants. Equation (3) defines the output of each control system as a "leaky integrator". This is done by accumulating a proportion, s, of the change in output, g \((S - P) - O\), that occurs during each time interval of the simulation, dt, into the current output, O. Representing output as the time integral of error seemed appropriate for a model of prescription writing because the actual prescription text — the output produced by the physician writing a prescription — is the time integral of the thoughts and actions that produce that text over time.

**Closed loop control**

Equations (1) through (3) define a closed loop control system. Since the output of equation (1) is an input to equation (2) and the output of equation (2) is an input to equation (3) and the output of equation (3) is an input to equation (1), there are circular references in the spreadsheet, which are resolved by iterative calculation. Each iteration of the model represents the passage of one unit of time, \(dt\), which was set to correspond to 1 sec of real time. The parameters of the model are set so that each control system is dynamically stable and brings the prescription component it controls to the intended (specified) state in a relatively short time, about 10 to 20 iterations (sec).
Since prescription components would be written in sequence in a real prescribing situation, the model takes the equivalent of about one minute of real time, on average, to write a prescription. One study of prescription writing found that the time taken to write a prescription in inpatient environments is about 5 minutes (Tierney et al, 1993). The prescribing model is set up to write prescriptions in about 1 minute because the model does not perform some of the time consuming activities, such as producing multiple prescription entries, that were included in the measurements of the time to do inpatient prescribing. Still, it takes considerably less than 1 minute to simply write a prescription. The model was set up to take approximately 1 minute to produce a prescription in order to represent the time taken by the thought processes involved in formulating a valid prescription before each prescription component is actually written down.

**Skill and Agility**

The constants in equation (3), g and s, which are defined in rows g and s of the spreadsheet (see Figure 2), determine how well and how quickly each control system writes a prescription component. The constant g is the **gain** of the control system. The higher the gain of a control system (the larger the value of g) the better the system controls. Better control means that the system is able to keep the controlled variable (the prescription component, q, in this case) close to the intended value, S, protected from the effects of disturbance. So gain corresponds to the **skill** of the physician writing the prescription. The constant s determines how quickly the system acts to bring the controlled variable to the intended value. The larger the value of s, the greater the speed of response of the control system. So s corresponds to the **agility** of the physician writing the prescriptions.

Gain and speed are basically independent parameters of control system operation, although the speed of response does place limits on gain: The maximum gain of a high-speed system is lower than that of a low speed system (Powers, 1979). Nevertheless, a high gain (highly skilled) system can act quickly or slowly within the limits placed on gain by speed. A high gain system will bring the controlled variable close to its intended value, but the high-speed version of this system (with large s) will achieve this more quickly than the low speed version of the same system (with small s).
5. PRESCRIPTION PERCEPTION

The prescription-writing model represents the components of a prescription order, q, as perceptual variables, P, (Powers, 1973). The process that produces a perception of each component is similar to the process assumed by receptive field models of perception. These models come out of the single cell physiological studies of Hubel and Weisel (1968). In receptive field models, sensory input, such as the image of a prescription component, is mapped, via neural network computations, into the firing rate of a single neuron. The rate of neural firing is, thus, a perceptual representation of the state of the sensory input. Different firing rates represent different states of the sensory input. For example, if the sensory input (the image of the prescription component being written on the prescription pad) is a drug name, then one firing rate (one value of P) might indicate that the sensory input is "dicloxacillin" while another firing rate might indicate that the sensory input is "doxycycline". As firing rate approaches the value that corresponds to "dicloxacillin", the perception of what is being written approaches the drug name "dicloxacillin". Similarly, as firing rate approaches the value that corresponds to "doxycycline", the perception of what is being written approaches the drug name "doxycycline".

The receptive field approach to perception is a natural choice for a control model, which acts on the basis of the quantitative difference between two scalar variables, S and P. Multidimensional sensory inputs, such as baseball trajectories (Marken, 2001), can be controlled only if they are represented as time varying scalar quantities. Therefore, a simplified version of the receptive field function was implemented in the prescription-writing model. The prescription component that is controlled by each system was represented in the model as a single value rather than as a vector and the neural network perceptual function was represented as a simple multiplication of q by a constant, as shown in equation (2). Nevertheless, this simplified version of the model accurately represents the significant functional characteristics of the receptive field approach to perception.
6. WRITING A PRESCRIPTION

The control model writes a prescription by bringing the perception, $P$, of each prescription component to its intended state, $S$. Each prescription is made up of a different set of randomly selected values of $S$, which represent the specifications for the Drug, Dose, Route and Other components of that particular prescription. The $S$ values for each prescription are drawn randomly from a range of values, the width of which is a parameter specified by the user of the model. The control system model iterates through the calculations represented by equations (1) through (3) to bring the value of $P$ for each prescription component as close as possible to its intended (specified) value, $S$.

The prescription-writing behavior of a system controlling one prescription component is shown in Figure 3. System variables are shown changing over a 20 sec time period. At the start of the prescription-writing process the prescription component being written, $q$, differs considerably from its intended value, $S$, so that error $(S - P)$ is large. As the model writes the prescription, $q$ is brought to the intended state, $S$, and error exponentially approaches, but does not necessarily become equal to zero. How close error eventually comes to zero depends on the system's gain, $g$. The rate at which error approaches zero also depends on system gain as well as speed, $s$, and the dynamics of the disturbance affecting $q$ while the prescription is being written.

![Figure 3. Behavior of error (S-P), prescription component (q) and component specification (S) over time.](image-url)
Prescription completion rule

The prescription control process shown in Figure 3 would continue forever if it were not stopped when the prescription is considered complete. It was necessary, therefore, to provide the model with a rule for deciding that a prescription is complete and the prescription-writing control process should stop. One approach is to consider a prescription complete when error (S-P) equals zero. When the error in all prescription component control systems goes to zero the prescription being written is in the intended state and it can be considered successfully completed. This approach fails, however, because error rarely equals exactly zero. This means that many prescriptions are never counted as completed no matter how much time passes, something that never actually happens in prescription writing.

Another possible prescription completion rule is to consider a prescription component complete when the error in all prescription component control systems is less than some criterion value. So error doesn’t have to be exactly zero, just within some band around zero. It was found, however, that the behavior of the model using this error size completion rule was unstable and did not produce results that matched the available data.

The approach that was used to decide that a prescription is complete was to base this decision on the amount of time spent by each prescription component control system writing the component. A prescription component can be considered complete once a fixed amount of time has elapsed. The elapsed time completion rule is indicated by the vertical bar in Figure 3. The behavior of the model using this rule was found to be stable and could be made to match the available data quite well. The elapsed time completion rule also makes sense in terms of actual physician behavior. Producing a prescription is just one of several time critical tasks that a physician must perform. The amount of time a physician can spend writing a particular prescription is limited and varies from one prescription to the next. The model physician, therefore, considers a prescription complete when all prescription components are close to their specified value and it is time to go on to other activities (the time to complete the prescription has expired).

The time available for writing each prescription component was represented in the model as a uniformly distributed random variable with a minimum value corresponding to the least amount of time the physician would spend writing a prescription component before going on to another activity. In the simulation runs it was found that allowing an average time of 17 seconds for writing each prescription component, with a
minimum time of 11 seconds, gave the best fit to the available prescription error data. As soon as the time available for writing all prescription components expired the prescription was considered complete.

Prescription writing, therefore, consists of all four control systems doing what is shown in Figure 3: producing outputs (both thoughts and actions) in order to bring a perception, \( P \), of a prescription component, \( q \), to the intended state, \( S \). The model writes thousands of different prescriptions (represented by different sets of \( S \) values) in just a few seconds of real time. Each prescription is written using a different randomly selected time limit for deciding that all prescription components are complete. When the system is unable to bring a prescription component satisfactorily close to the intended state before the completion time cut off the prescription is considered to be in error. The model keeps track of the rate at which such prescription errors occur.

**Speed and Accuracy**

The model bears a family resemblance to control models of the speed-accuracy trade off (Connelly, 1984). Accuracy, in terms of prescription error rate, decreases as the speed of prescription writing, in terms of the time available for prescription completion, increases. The prescription-writing model differs from other models of the speed-accuracy trade, however, in terms of the source of limits on the time available for producing behavior. Models of the speed-accuracy trade apply to situations where these time limits are one of the goals of the actor. The model of prescription writing applies to situations where these time limits are imposed by task demands, such as the need to attend to another patient.

7. **CONTROL ERROR AND PERFORMANCE ERROR**

The error that drives the actions of the prescription-writing control model is not the same as the error seen when a physician writes an incorrect prescription. The error that drives the control system can be called control error while the error that is seen when a physician writes an incorrect prescription can be called performance error. Control error is the time varying difference between the intended and perceived state of the prescription \( (S - P) \), as shown in Figure 3. Performance error is based on the judgment of an expert, such as a
pharmacist or another physician, that there is something wrong with one or more of the completed prescription components, q. Prescribing errors are performance errors.

**Defining a prescribing error**

Because the model considers a prescription complete when a certain amount of time has elapsed, not when control error is zero, the model will often produce completed prescriptions when there is still control error in the systems controlling each prescription component. It was assumed that most of these completed prescriptions would be judged to be correct by an expert observer even though there is still control error in the system writing the prescription. Therefore, the decision regarding the correctness of a completed prescription (whether or not a prescription error has occurred) must be based on something other than the existence of control error. The approach taken was to assume that experts base their decisions about the correctness of a prescription on a comparison of a **categorical** representation of the state of each completed prescription component, q, to a categorical representation of the specification for the correct state of that component, S.

A categorical representation of each prescription component, q, was produced by rounding the real value of each prescription component to an integer. Since the specifications for each prescription component, S, were already integers, decisions regarding the correctness of a completed prescription were made by comparing the integer value of each prescription component to its integer specification. The result of comparing the rounded value of a prescription component, q, to the integer specification for that component, S, was either an exact match, meaning that the prescription is correct (no prescription error), or a mismatch, meaning that the prescription is in error. This decision procedure seems appropriate because it resulted in many prescriptions being judged correct even though there was still control error in the systems that had written these prescriptions.

**Categorical decisions**

Rounding the value of a prescription component, q, to an integer is equivalent to saying that each component will be seen categorically: as one thing or another, not something in between. If, for example, the actual prescription component has a value of 4.3, it is between two integer specifications (possible values of S). If the prescription component is a drug name and one intended value of that name is "Atrovent" and the other is "Alupent", then rounding q is equivalent to
assuming that the written prescription component will be seen as either "Atrovent" (corresponding to S=4) or "Alupent" (corresponding to S=5) but not as something in between, like "Aluvent" (corresponding to q = 4.3). Using the rounded value of q (4 in this case) to represent the prescription component is equivalent to assuming that this component will be seen as "Atrovent".

The method used to decide whether or not a prescription is in error assumes that the expert knows what prescription order the physician intended to write. That is, the method assumes that the expert knows the values of S that are used to write each prescription. This seems to be the case in actual studies of prescription error (Leape et al, 1995; Lesar et al, 1997). Experts appear to be able to determine, with considerable consistency, what the intention of the physician writing the prescription was or should have been. Indeed, experts would not be able to judge the correctness of a prescription if they were not able to infer the intentions of the physician who wrote the prescription.

8. ERROR RATE AND SKILL LEVEL

The behavior of the model was tested with several system parameters held constant: the average time to write a prescription, the magnitude of environmental disturbances that affect the state of the prescription order and the range of possible values of the specifications for each prescription component. Holding these parameters constant is equivalent to observing a physician who writes prescriptions in a particular practice, one that has a fixed workload (average time available to write a prescription), environment (disturbances of a fixed magnitude) and patient characteristics (which determine the range of specifications for prescriptions that are written). The speed, s, of all prescription component control systems was also held constant, at .007. All that was varied was the skill level of the model physician writing the prescriptions. Skill level was varied by varying the gain, g, of the systems writing the four prescription components. At each skill level, the systems controlling each of the four prescription components were set to the same gain value. The gain values ranged from 300 to 500. The error rate produced by the model as a function of skill level is shown in Figure 4.
The results in Figure 4 show that prescription error rate falls off exponentially as the model physician becomes more skillful at writing prescriptions. The parameters of the model were set so that the error rate produced at the highest skill level is in the range found in studies of prescribing error. The error rate at the highest skill level in Figure 4 (g = 500) is about 1%. This means that about 1 out of 100 prescriptions were considered to be in error because one or more components of these prescriptions were judged to be in error. This error rate is near the middle of the observed range of error rates (0.4 - 1.9%) that have been observed in studies of prescribing error.

The results in Figure 4 show how error can occur in highly skilled behavior. Most of the thousands of prescriptions that are written by the model at each skill level are correct. An error occurs when the model is unable to get a prescription component, q, to a state that is considered correct by an expert observer in an amount of time that is usually sufficient to accomplish this. Error is not caused by the occasional rushed prescription. Rather, it is caused by factors that keep the control system from bringing the prescription component to the specified state as quickly as usual.
Figure 4 shows error rate approaching zero as skill level increases. This suggests that it might be possible to eliminate prescribing error by producing perfectly skilled physicians. In fact, however, performance error rate never goes to zero, even when the model is operating at the highest skill (gain) levels. Performance error continues to occur at high skill levels because the control systems are unable to bring prescription components to their specified states instantly. The dynamics of the control process combined with the fact that there is an imposed (and unpredictable) limit to the amount of time available to write prescriptions means that, even if it were possible to significantly increase the skill (gain) of a highly trained physician, such a physician would still make performance errors, though at a very low rate.

It is also impossible to reduce performance error significantly by increasing the agility of the prescription writing control systems by increasing the speed parameter, $s$. If the agility (speed) of the prescription writing control systems is increased the maximum gain of these systems is reduced. Thus, while a highly agile system will bring the prescription component, $q$, quickly toward the reference specification, $S$, it will not be able to get it as close to $S$ as did the less agile system.

9. MODEL VALIDATION

Quantitative Validation

The validity of the model was tested by comparing the behavior of the model to data on the rate at which different types of prescribing errors occur (Leape et al, 1995; Lesar et al, 1997). These studies determined the number of times a prescribing error was the result of writing an incorrect drug name, dosage, route, or other aspect of a prescription. The results of these studies were combined and the observed rates of different prescribing error types are shown in the top row of Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Drug</th>
<th>Dose</th>
<th>Route</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leape/Lesar Data</td>
<td>39%</td>
<td>57%</td>
<td>3%</td>
<td>1%</td>
</tr>
<tr>
<td>Model Data</td>
<td>39%</td>
<td>57%</td>
<td>4%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 1. Observed rate of different types of prescription error (top row) and rates of the same types of error produced by prescription-writing model.
The model was fit to the data in Table 1 by adjusting only the speed parameter, s, for each prescription component control system. The gain of all the systems was held constant at 500. This is equivalent to assuming that the model physician is equally (and highly) skilled at writing the Drug, Dosage, Route and Other components of a prescription. Other model parameters — the distribution of times available to write each component, disturbance magnitude and the range of specifications, S, for each component — were also held constant for all four systems, which is equivalent to assuming that the model physician writes all prescription components under nearly the same circumstances. The results of one model run are shown in the lower row of Table 1.

The results in Table 1 show that the distribution of rates of different error types produced by the model corresponds almost exactly to the empirical distribution of these rates. The values of s that produced these results were 0.000684, 0.000669, 0.000731 and 0.000738 for the Drug, Dosage, Route and Other component writing control systems, respectively. The larger s the faster the actions of a control system so the system writing the Dosage component was slowest while the system writing the Other component was fastest. Further quantitative validation of the model comes from the fact that the model's distribution of rates of different error types was produced with an overall error rate of 1%, which is close to the overall error rate found in the empirical studies of the rate of different error types (Lesar et al, 1997).

**Qualitative Validation**

The prescription-writing model suggests that prescribing error rates will be highest in situations where the time available to complete prescriptions is limited. Though there is no quantitative evidence regarding the relationship between time taken to write prescriptions and prescribing error rate, there is qualitative evidence that distractions and workload, factors that affect the time available to write prescriptions, are the major factors contributing to the occurrence of prescribing error (MedMaRx, 1999). In the prescription-writing model, shortening the average time to decide that a prescription is complete increases the probability of a prescribing error. The fact that the same thing seems to happen (as a result of distractions and workload) when physicians write prescriptions provides some qualitative evidence for the validity of the prescribing model.
10. MODEL EXCURSIONS

The model was used to determine how factors other than skill influence error rate. These factors were studied by varying the main parameters of the model while skill level was held constant at 500 for all systems. This "highly skilled" model was then run to see how excursions in 1) the average time to write a prescription 2) the magnitude of environmental disturbances and 3) the range of possible values of the specifications for each prescription component affected the overall error rate.

Time Available to Write a Prescription

The effect of changes in the average time available to write a prescription on error rate was substantial. Cutting the average time to write a prescription component in half (from 17 to 8.5 seconds) produces a 50-fold increase in error rate (from 1% to about 50%). Doubling the average time to write a prescription component (from 17 to 34 seconds) reduces error rate to nearly zero. The enormous effect of timing on error rate is not surprising because control error decreases exponentially over time (see Figure 3). When the time available to complete a prescription component is short, the difference between q and S is likely to be relatively large when the prescription is completed. Shortening the time available to write a prescription exponentially increases the chances that the completed prescription will be classified as a prescribing error. Similarly, lengthening the time available to write a prescription exponentially decreases the chances that the completed prescription will be classified as a prescribing error.

Disturbance Magnitude

Environmental disturbances are design features of the prescription-writing system (pens, pads, drug name similarities, etc.) that can interfere with the ability to produce correct prescription components. The magnitude of these disturbances corresponds to the impact of poor system design characteristics on the prescription components, q, that are being written. The model excursions suggest that, below a certain magnitude, these disturbances have surprisingly little effect on prescribing error rate. Doubling the magnitude of environmental disturbances did result in a five-fold increase in prescription error rate (from 1% to about 5%). However, halving the magnitude of these disturbances resulted in virtually no decrease at all in error rate.
The results of the excursions with disturbance magnitude suggest that there is little to be gained, in terms of error rate reduction, by removing "error causing" events from the environment when the errors that do occur are being produced by a highly skilled control system. The fact that actual prescribing error rates are quite low indicates that physicians are, indeed, highly skilled prescription-writing control systems. Physicians are, therefore, already effectively protecting the variables they are controlling from prevailing disturbances. Further reductions in the magnitude of disturbances have no effect on error rate because these disturbances were having little or no effect on error rate before the disturbance reductions were put into effect. Improvements in system design are, therefore, not likely to produce significant reductions in error rates when error rates are already very low.

Range of Prescriptions

The range of possible specifications for each prescription component, S, corresponds to the range of different prescriptions that a physician might write in a practice. Doubling this range increased error rate about 7 fold (from 1% to 7%). Similarly, halving this range reduced error rate nearly to zero. As in the case of the time available to write a prescription, the effect of the range of prescriptions is related to the exponential falloff of control error (Figure 3). When the range of prescriptions is large there will tend to be a large change, from one prescription-writing trial to the next, in the value of the specification sent to each system. This means that, on each trial, the prescription-writing control systems will tend to start with a large initial control error (as seen in the error curve in Figure 3). The systems take a longer time to reduce error that starts large (as it does when the range of specifications is large) than error that starts small (as it does when the range of specifications is small). So prescriptions written when the range of prescriptions is large are more likely to be completed with the difference between q and S still relatively large than are prescriptions written when the range of prescriptions is small.

11. TYPES OF PRESCRIBING ERROR

Slips

The prescribing model described in this paper is an attempt to mimic the behavior of a skilled and informed physician. The prescribing
model’s intentions, \( S \) (that is, the prescription components the model intends to write) are assumed to be correct with respect to the clinical situation. The errors committed by the prescribing model are what error theorists call slips (Norman, 1990). A slip occurs when the result produced by an actor is not what was intended. For example, a slip occurs when a physician intends to write “100 mg” but writes “10 mg” instead. The prescribing model commits a slip when a result produced by the model – a prescription component, \( q \) – is not what was intended, \( S \). In fact, however, a large proportion of actual prescription errors are what error theorists call mistakes.

**Mistakes**

A mistake occurs when the result produced by an actor is what was intended but the intention itself was wrong. For example, a mistake occurs when a physician intends to and successfully writes “10 mg” when the intention should have been to write “100 mg”. Prescribing mistakes are errors that result from incorrectly applied knowledge (Lesar, 1997). For example, mistakes occur when the physician incorrectly recalls the appropriate route for administration of a particular drug or forgets to check the patient’s record for possible interacting drugs or allergies. Such knowledge slips lead the physician to adopt the wrong intention for a prescription. So the physician prescribes intravenous administration of a drug that should be taken orally or prescribes aspirin for a patient with a documented aspirin allergy. These kinds of mistakes can be viewed as slips that are made by higher level control systems, the one’s that set the intentions (prescription component specifications, \( S \)) that are carried out by the systems described in the prescribing model.

Looking at mistakes as slips made by higher level control systems is based on a hierarchical control model of prescribing, with higher level control systems acting by setting the specifications, \( S \), for the variables controlled by lower level systems (Marken, 1990). The higher level systems make mistakes for the same reason that lower level systems make slips: they decide that the control process is finished before control error is sufficiently reduced. Thus, a control model of prescribing can explain, at least in principle, both slips and mistakes. The overall error rate produced by the prescribing model described in this paper can be thought of as including both of these types of error. However, a hierarchical version of the prescribing model is needed to account for the relative proportion of slips and mistakes that actually occur in prescription writing.
Failures

A third kind of prescribing error occurs when the physician has no access to the information necessary to write a correct prescription. For example, the physician may have no way of knowing what potentially interactive drugs the patient is taking or what drug allergies the patient might have. This information might not be available from the patient or medical records. When the physician prescribes a drug that interacts with drugs the patient is taking or to which the patient is allergic, an error is committed that is neither a slip nor a mistake because the physician could not possibly have avoided it; the necessary information was just not available. These kinds of errors can be called failures. The wrong prescription was written and there was nothing the physician could have done to prevent it. The prescribing model described in this paper does not make failure errors.

12. ERROR PREVENTION

The goal of developing the model of prescribing error was to learn how such errors might be prevented. The model excursions suggest that some approaches to prescribing error prevention may be more effective than others. For example, one possible approach to prescribing error prevention is to increase physician skill and/or agility through training. However, the model suggests that increases in skill or agility will have little effect on prescribing error rates. This is because skill and agility are not what is driving prescribing error rates in the model. Once the model physician reaches a very high level of prescription writing skill and agility (a level that results in error rates on the order of 1%) the main driver of prescribing error rates is the time limit imposed on prescription production.

Another approach to prescribing error prevention would be to reduce environmental disturbances, such as poor system design features, which can cause error. However, the model suggests that the magnitude of existing environmental disturbances is well within the limits of what can be handled by physicians writing prescriptions. Further reduction of such disturbances would probably have very little effect on prescribing error rate.
Reducing the range of prescriptions

The prescribing model suggests that one effective approach to preventing prescribing errors would be to reduce the range of different types of prescriptions written by a physician. However, though this approach might be effective, it is probably not feasible. The range of different types of prescriptions written by a physician is shaped by the nature of the physician’s practice, not by the physician.

Changing Work and Workflow

One feasible approach to error prevention suggested by the model is to increase the time available to the physician for writing a prescription. This can be achieved by reducing distractions and workload. Both distractions and workload can be reduced by changing workflow - the way work is done. One approach to changing workflow involves requiring that prescriptions be entered into an electronic database called a physician order entry (POE) system. Studies of POE systems show that the use of such systems reduces error rate considerably while doubling the time it takes to write a prescription (Tierney, et al, 1993). The prescription-writing model described in this paper suggests that the error rate reduction found with POE systems may actually be the result of the increased time physicians must spend producing each prescription.

Electronic prescribing systems

The prescribing model was used to test two other possible approaches to error reduction based on the use of electronic prescription support systems (e-prescribing systems). One approach was based on the use of an electronic decision support system that would catch prescribing errors with a certain probability. This system was added to the model by having a simulated decision support system detect, with some probability, prescriptions that would be counted as errors if they were completed. When such errors were detected prior to prescription completion the model would write a new prescription. So there was still some small chance that the model would make another error that would go undetected. The results of running the model with this decision-support error reduction scheme were not surprising: error rate was reduced in proportion to the probability that the decision support system detects errors. If the decision support system detects errors with a probability of .5 then the prescribing error rate is cut in half. So if it is possible to design a system that will detect prescribing errors with some probability then prescribing error can be reduced in proportion to the probability that the system detects errors.
Another approach to error reduction based on the use of e-prescribing systems was based on the use of a system that would fill out the remaining components of a prescription once one component, such as the drug name, was specified. In the model, this was tested by reducing the number of systems writing prescription components from four to one. Reducing the number of prescription components to be written should reduce the chances that an error in one component will result in a prescription being classed as an error. And, indeed, reducing the number of prescription component writing systems from four to one cuts the prescription error rate in half.

These model-based estimates of the potential benefits of e-prescribing systems rest on the unlikely assumption that the use of such systems will not introduce errors into the prescribing process. The prescribing model implementation of e-prescribing systems assumes that these systems can only help, not hinder, the prescription-writing process. In fact, e-prescribing systems probably lead to their own as yet undetermined rate of prescribing errors, which may wash out any reduction in error rate that might result from the use of such systems.

13. DISCUSSION

The prescription-writing model shows that prescribing errors are not the fault of the physician or the environment. Prescribing error results from the fact that perfect performance is impossible, even by a highly skilled control system, such as a physician, when there is a limit to the time available to produce a prescription. There is always some chance that a prescription will be considered to have been correctly completed by the person writing the prescription when, in fact, it was not. The chances of such errors occurring can be reduced to a very low level but they cannot be eliminated completely unless an unlimited amount of time were always available to the physician for completing each prescription. But even the rare error can produce catastrophic consequences when the error is an incorrect prescription for medication. Since it is impossible to completely eliminate the occurrence of such errors it is probably wise to follow the advice of Rasmussen (1986) and try to design the prescribing process so that the negative consequences of error are minimized. This might be accomplished by improved monitoring of patients taking potentially dangerous
medications or by increasing the checks that prescriptions for such medications must pass before administration.

The prescription-writing model (and much of the prescription error literature) deals only with errors of commission: prescriptions that have been written and are incorrect. A far more prevalent source of prescription errors may be errors of omission: prescriptions that should have been written but were not. It is difficult to estimate the rate at which such omission errors occur but some estimates go as high as 60%: six out of 10 patients fail to receive prescriptions for medication they need. This suggests that there may be more to be gained from looking at ways to reduce prescription omission errors than from looking at ways to reduce the already small number of prescription commission errors.
REFERENCES


