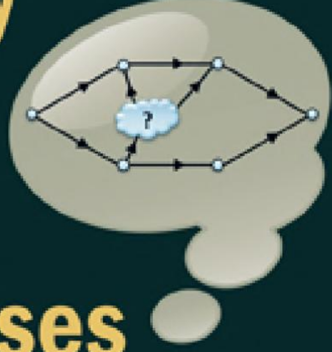
Advanced Series on Mathematical Psychology Vol. 4

Discovering Cognitive Architecture by Selectively Influencing Mental Processes

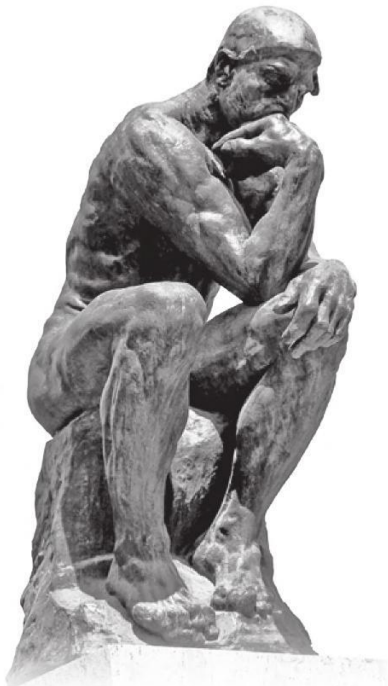


Richard Schweickert
Donald Fisher
Kyongje Sung

World Scientific

Advanced Series on Mathematical Psychology Vol. 4

Discovering Cognitive Architecture by Selectively Influencing Mental Processes



Richard Schweickert
Purdue University, USA

Donald L. Fisher
University of Massachusetts Amherst, USA

Kyongje Sung
Johns Hopkins University
School of Medicine, USA

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

Library of Congress Cataloging-in-Publication Data
Schweickert, Richard.
Discovering cognitive architecture by selectively influencing mental processes / by Richard Schweickert, Donald L. Fisher & Kyongje Sung.
p. cm. -- (Advanced series on mathematical psychology ; v. 4)
Includes bibliographical references and index.
ISBN 978-981-4277-45-7 (hardcover : alk. paper)
ISBN 981-4277-45-2 (hardcover : alk. paper)
1. Psychology--Mathematical models. 2. Psychometrics. I. Fisher, Donald L. II. Sung, Kyongje. III. Title.
BF39.S345 2012
150.1'5195--dc23

2012002594

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

Printed in Singapore.

Contents

This page intentionally left blank

Chapter 1

Introduction to Techniques

A person performing a task such as searching a screen for a target executes mental processes such as perceiving, recognizing, selecting a response and so on. In the early days of experimental psychology Wundt tried to directly find the duration of a single process, apperception, by asking an observer to directly insert this process into a task or remove it. For Wundt, perception denoted "the appearance of a content in consciousness" after a stimulus is presented, and apperception denoted a deeper process, "its reception into the state of attention" (Külpe, 1895, p. 426). Investigators in Wundt's lab presented stimuli to observers trained in introspection. In one condition the observer was instructed to respond when the stimulus was perceived, and in another condition the observer was instructed to respond when the stimulus was apperceived. By subtracting the reaction time for the perception condition from the reaction time for the apperception condition, the time for apperception itself could be found. It would have been fortunate if this naive procedure had worked, but even at the time it was unconvincing. Cattell (1893) said that "the great variation ... of the measurements bears witness to the lack of an objective criterion."

The approach of Donders (1868) to inserting processes was more objective. The assumption was that processes required for performing a task are executed one after the other, in series, and the reaction time is the sum of the times required for the individual processes. The aim was to insert or remove processes by changing the task to be done. For example, in one experiment, the stimulus was a vowel sound and the observer's response was to repeat the vowel. In the first condition (a), the simple condition, the observer knew which vowel was to be presented and only had to repeat it. In a second condition (b) the observer did not know which of five vowels was to be presented, and had to repeat whichever was presented. Condition (b) requires two processes in addition to those required in condition (a), namely, (1) the stimulus vowel must be discriminated and (2) the response vowel must be chosen. Thus, subtracting reaction time for (a) from reaction time for (b) gives the time required for discrimination plus the time required for choice.

In a third condition (c) the observer did not know which of five vowels was to be presented, but only had to respond to one, say, *i*, by repeating it when it was presented. No choice was required for the response, although discrimination of the stimulus vowel was required. Thus, subtracting response time for (b) from response time for (c) gives the time required for choice. Subtracting response time for (a) from response time for (c) gives the time required for discrimination.

Donders reported that the mean reaction time for condition (a) was 201 msec, for (b) 284 msec, and for (c) 237 msec. Then the time for discrimination is c − a = 36 msec, and the time for choice is b − c = 47 msec.

An important feature of this *Subtractive Method* of Donders is that the experimenter can determine, based on the observer's responses, whether the intended task was performed. Nonetheless, the experimenter does not know how the task was performed. A criticism at the time, based on introspection, was that changing from one condition to another changes the nature of the processing, even of processing that precedes the process allegedly inserted (Külpe, 1895, p. 414).

In principle, the results of Donders' subtractive method can be checked. To take a simple case, suppose the experimenter can insert and remove two processes, with durations x and y. By producing three tasks intended to have reaction times, respectively, of

$$x$$

$$y$$

$$x + y$$

the experimenter can check whether the duration of the third task is indeed the sum of the durations of the first two tasks. This example is an over simplification, but with three or more processes, if the experimenter inserts and deletes processes in the right combinations, the number of observed response times can be made large enough so that values of unknown process durations can be solved for in more than one way, allowing a check. Curiously, this does not seem to have been done at the time.

Stretching Processes Rather Than Inserting Them

In Sternberg's (1969) elegant approach, instead of trying to insert a process, the experimenter tries to change the task slightly, in order to make an existing process take longer, without changing anything else. Such a manipulation is called a factor, and is said to *selectively influence* the process.

The two major assumptions of the theory are (1) processes are executed in series, so the reaction time is the sum of the durations of the individual processes, and (2) each of two experimental factors prolongs a different process. There are also secondary assumptions about the measurement of time and so on. These are numerous and so minor we can safely assume they are met. Then the theory predicts the combined effect of prolonging two processes will be the sum of the effects of prolonging them individually.

This prediction can be tested with an Analysis of Variance (ANOVA). Factors having additive effects on response time are called *additive factors*. Two factors that are not additive are said to *interact*. It is common to call processes executed one after the other *stages*. An experiment with additive factors supports the theory. If two factors interact, at least one of the major assumptions is wrong. Sternberg (1969) proposed that if two factors interact, it is likely that assumption (2) is violated and the two factors influence the same stage.

The technique of selective influence for a series of stages is called the *Additive Factor Method*. With it, the experimenter obtains an immediate check on the assumptions through the test of interaction in the Analysis of Variance. Its applications have been numerous, see, *e.g.*, Sanders (1990) and Sternberg (1998).

As a well-known example, in experiments of Sternberg (1966, 1967), the subject was given a set of digits to memorize, the *positive set*. On each trial a digit was presented, and the task was to respond whether the presented digit is in the positive set or not. The task is now called the *Sternberg memory scanning task*, or the *memory search task*. Two factors discussed by Sternberg (1969) are (1) a change in stimulus quality produced by superimposing a checkerboard pattern on the stimulus digit and (2) a change in the size of the positive set from 1 to 2 to 4 digits. The two factors had additive effects on reaction time (Sternberg, 1966, 1967). The interpretation is that the first factor selectively influences one stage, stimulus encoding, and the second factor selectively influences another stage, memory comparison; further, the two stages are arranged in series. Other stages in the series were selectively influenced by other factors, see Sternberg (1969).

When the combined effect of two factors is the sum of their separate effects, we say the composition rule is addition. In that case, there is a model of the situation in which two processes are in series, with each factor selectively influencing a different process.

However, additivity of the factors does not imply that two processes in series exist in reality, because other process arrangements could yield an additive composition rule. Tension between what can be observed and what can be inferred has been part of cognitive psychology since its inception, because the subject matter, cognition, is only partly observable. Consider the case of two factors, Factor $\mathcal{A}$ and Factor $\mathcal{B}$, having additive effects on reaction time. Suppose each factor has two levels. Let τ_{11} be the reaction time when both factors are at level 1, let τ_{12} be the reaction time when Factor $\mathcal{A}$ is at level 1 and Factor $\mathcal{B}$ is at level 2, and so on. The two factors have additive effects if changing the level of Factor $\mathcal{A}$ from 1 to 2 has the same effect at each level of Factor $\mathcal{B}$. That is,

$$\tau_{21} - \tau_{11} = \tau_{22} - \tau_{12}.$$

To construct a model with two processes in series, let process A have duration

$a_1 = .5\tau_{11}$ when Factor $\mathcal{A}$ is at level 1

$a_2 = .5\tau_{11} + \tau_{21} - \tau_{11}$ when Factor $\mathcal{A}$ is at level 2.

Let process B have duration

$b_1 = .5\tau_{11}$ when Factor $\mathcal{B}$ is at level 1

$b_2 = .5\tau_{11} + \tau_{12} - \tau_{11}$ when Factor $\mathcal{B}$ is at level 2.

Finally, suppose the response time for a level of Factor $\mathcal{A}$ combined with a level of Factor $\mathcal{B}$ is the sum of the corresponding durations of process A and process B.

It is easy to check that when both factors are at level 1, the reaction time is τ_{11}, when Factor $\mathcal{A}$ is at level 1 and Factor $\mathcal{B}$ is at level 2, the reaction time is τ_{12}, and so on. Each factor changes the duration of only one process. Hence, the data can be represented by two processes in series, with durations as above, and with each factor selectively influencing a different process. (See Dzhafarov and Schweickert, 1995, for a representation in which the reaction times and process durations are random variables, rather than fixed constants.)

Clearly, it is arbitrary to use $.5\tau_{11}$ in the durations above, so this is not the only way to represent the data with two processes in series. More troubling is that a quite different process arrangement can also represent the data. We have implicitly assumed that a process begins processing at a starting point and stops processing at a finishing point, and if a second process follows, its starting point is the finishing point of the first process. McClelland (1979) and Townsend and Ashby (1983) showed that factors can have additive effects on reaction time in a different kind of model, where as soon as a process begins, it starts sending output to its successor. McClelland's model is called the cascade model, and Eriksen and Schultz (1979) call such models continuous flow models. An analysis of the cascade model by Roberts and Sternberg (1993) showed that it failed to account for aspects of their data. But it often happens that two process arrangements account for the known data equally well. In the end, the choice between them can only be based on nonempirical considerations such as simplicity, plausibility and taste.

If two factors do not have additive effects on reaction time, it is possible that each factor prolongs a different process, so assumption (2) above is satisfied, but the processes are not in series, so assumption (1) is violated. Sternberg (1969) pointed out that if processes are in parallel, the effect of prolonging two of them would be the maximum of the effects of prolonging them separately.

In some situations, the processes are not all in series and they are not all in parallel. Evidence comes from dual tasks, in which two stimuli are presented, and a response is made to each (Telford, 1931). Consider the case of two stimuli presented at the same time. When results are compared with the corresponding single tasks in which each reaction is made separately, the following outcomes are typical (*e.g.*, Schvaneveldt, 1969), although not always found. In the dual task, the subject is not carrying out all the single task processing for the first response, followed by all the single task processing for the second response, because the time required to do the dual task is less than the sum of the times required to respond to each stimulus separately. On the other hand, in the dual task the subject is not carrying out all the single task processing for the first response simultaneously with all the single task processing for the second response, because the time to do the dual task is longer than the maximum of the times required to respond to each stimulus separately. Something more general than pure serial or pure parallel processing is needed.

Chapter 2

Introduction to Process Schedules

The main reason for selectively influencing processes is to learn about the arrangement of the processes in a structure containing them. It is clear that there may not be a single structure used for all tasks. Meyer and Kieras (1997a, 1997b) emphasize that a system with flexible strategies will operate in a variety of ways. This chapter introduces two structures, task networks and trees, which are plausible, tractable and testable. The former are often used for modeling reaction times, the latter for response probabilities. Other structures will be introduced in later chapters.

Gantt Charts and Directed Acyclic Task Networks

Bar charts are a natural way to represent the mental processes required for a task; they are especially useful when intuition about process durations is important. Bar charts are also called Gantt charts. Figure 2.1 gives an example for processes in a dual task. Stimulus $s1$ is presented, followed after a stimulus onset asynchrony (SOA) by stimulus $s2$. Responses $r1$ and $r2$ are made to stimuli $s1$ and $s2$ respectively. There are three sequential processes for each stimulus, a perceptual process, A, a central process, B, and a motor preparation process, C. (A motor movement follows motor preparation, but because reaction time is ordinarily measured at response onset, the motor movement that follows is ordinarily not illustrated.) In this model, the perceptual processes for the two stimuli, $A1$ and $A2$, are executed concurrently. However, the response to the second stimulus is delayed because, in accord with Welford's (1952, 1967) single channel theory, the central processing, $B2$, for the second stimulus cannot begin until the central processing, $B1$, for the first stimulus is finished. The central processes $B1$ and $B2$ are executed sequentially. The first use of this model that we are aware of is by Davis (1957); it was popularized by Pashler and Johnston (1989). For more discussion, see Pashler (1994).

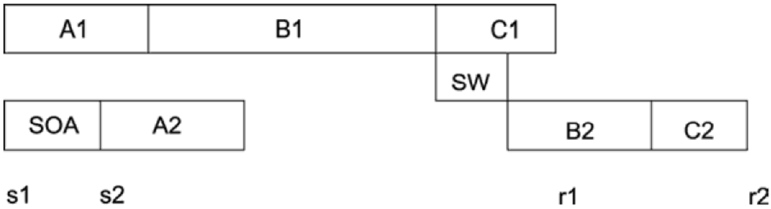


Fig. 2.1. Gantt chart for a dual task.

When intuition about relationships among processes is important, a Gantt chart is often replaced with an equivalent directed acyclic task network. Figure 2.2 shows a directed acyclic task network corresponding to the Gantt chart in Figure 2.1. The network is directed because each arc has a direction, and it is acyclic because no process precedes itself; that is, one cannot go from the head of an arc to its tail by following a sequence of arcs, each from tail to head. A wide variety of models are explicitly or implicitly in the form of Gantt charts or directed acyclic task networks. These include serial models (Donders, 1868; Sternberg, 1969), parallel models (Townsend, 1972), and the dual task model already mentioned (Davis, 1957; Pashler & Johnston, 1989). They also include models of de Jong (1993); Ehrenstein, Schweickert, Choi and Proctor (1997); Fisher and Glaser (1996); Johnston, McCann and Remington (1995); Osman and Moore (1993); Pashler (1984); Ruthruff, Miller, and Lachman (1995); Van Selst and Jolicoeur (1994); and Welford (1952). The various models make different predictions about details, but because they all can be represented as Gantt charts (or, equivalently, as directed acyclic task networks), there are certain general predictions they all make. If one of the general predictions fails for an experiment, there is no possible directed acyclic task network in which the experimental factors selectively influence different processes.

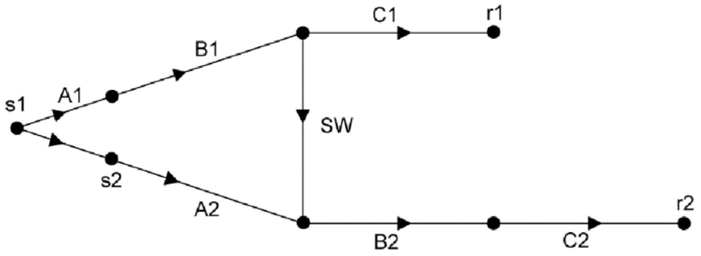


Fig. 2.2. Directed acyclic network equivalent to Gantt chart for dual task.

If the processes in a task cannot be represented in an acyclic task network, they can sometimes be represented in a more general structure, an OP (Order-of-Processing) diagram. These were introduced by Fisher and Goldstein (Fisher and Goldstein, 1983; Goldstein and Fisher, 1991, 1992). They were first used to derive moments of response time distributions for task networks and other models. Later, the availability of expressions for the moments lead Fisher (1985) to propose the use of OP diagram representations for many different cognitive networks, such as queuing networks and Petri nets. These will be discussed in later chapters. For more background on the use of response times to analyze mental processes, the reader is referred to the excellent surveys by Luce (1986) and Townsend and Ashby (1983). For networks of queues, see Liu (1996), Miller (1993), and Wu and Liu (2008).

Directed Acyclic Task Networks

The directed acyclic task network in Figure 2.2 is made of vertices joined by arcs. Processing begins with the presentation of a stimulus at the starting vertex of the network. A mental process is represented by an arc directed from one vertex to another. The *starting vertex* of an arc, at the tail, represents the starting point of the process. The *ending vertex* of the arc, at the head, represents the finishing point of the process. Responses are made at the ending vertices of the network.

Sometimes an arc does not represent a mental process, but merely indicates that one process precedes another. For example, a stimulus onset asynchrony is represented by an arc directed from the onset of one stimulus, a vertex, to the onset of another stimulus, another vertex. This SOA arc does not represent a mental process. As another example, suppose a process stops using a certain resource at some point, represented by vertex, and at another point, represented by another vertex, a second process starts to use the resource. An arc from the first vertex to the second vertex can be used to represent the fact that the resource must be released by the first process before it can be used by the second. If the resource is available the instant it is released, the duration of the arc is 0. For convenience, we will often refer to arcs as processes, even when there is no processing going on. An arc with duration 0 representing precedence is called a *dummy process*.

By starting at a vertex and moving along arcs in the direction of their arrows until another vertex is reached, one traces a path. More precisely, a *path* from a vertex u to a vertex z consists of the vertex u, followed by an arc directed from u to a vertex v, followed by an arc directed from v to a vertex w, and so on, with the last arc having ending vertex z. A single vertex is considered a path. To indicate that one process immediately precedes another, the head of the arc representing the first process is incident with the tail of the arc representing the second. If one process precedes another (not necessarily immediately), there is a path from the head of the arc representing the first process to the tail of the arc representing the second; the path will go along arcs in the direction indicated by the arrows.

We say a vertex *precedes* another vertex if there is a path having at least one arc from the former vertex to the latter vertex. A process preceding a process, a vertex preceding a process, and so on are defined similarly. A path that goes from a vertex u to the same vertex u, and that has at least one arc, is called a *cycle*. An *acyclic network* has no cycles, so a vertex or process does not precede itself. We assume precedence is *transitive*, that is, if process *x* precedes process *y,* and process *y* precedes process *z*, then *x* precedes *z*.

Two processes are *sequential* or *ordered* if one precedes the other; otherwise they are *concurrent* or *unordered.* We use the term "concurrent" as in the operations research literature to mean "potentially concurrent." When we say two processes are concurrent, we mean there is no requirement for one of them to finish before the other can start. Typically, portions of their execution will overlap in time, but the processes might not literally be executed simultaneously and it is possible that one process would be completed before the other one starts.

Some processes begin execution as soon as the first stimulus is presented. These have their starting vertex at the starting vertex of the network. We assume every other vertex in the network represents an AND gate or an OR gate. A process whose starting vertex is at an *AND gate* begins execution as soon as *all* processes immediately preceding it finish. A process whose starting vertex is an *OR gate* begins execution as soon as *any* process immediately preceding it finishes. Some processes have their ending vertex at a response. The response is made as soon as *all* immediately preceding processes are finished if the response is at an AND gate, and as soon as *any* is finished if the response is at an OR gate.

In the networks considered here, except for the starting vertex, every vertex is an AND gate or every vertex is an OR gate. In the former case the network is called an *AND network* and the latter case an *OR network*. AND networks are often called *PERT* (Program Evaluation and Review Technique) networks or *critical path* networks (Kelley & Walker, 1959; Malcom, Roseboom, Clark & Fazar, 1959; Elmaghraby, 1977). For short, we will use the term *task network* to refer to an AND network or an OR network. Networks having both AND and OR gates, or other kinds of gates, are possible of course, but beyond the scope of this work.

Sometimes a task might appear to require both AND gates and OR gates, but closer analysis shows it does not. Consider a visual search task with a process working on each item on the screen, these processes being concurrent. Suppose on a target absent trial the response is made as soon as all of these processes finish, each with the answer "nontarget." Then the response "absent" is made at an AND gate. Suppose on a target present trial, several targets are present on the screen. Suppose the response "present" is made as soon as any process finishes with the answer "target." The processes working on the nontarget items can be ignored, because they will not trigger a response. Then the response "present" is made at an OR gate. At first it might seem that a single network with both an AND gate and an OR gate is required. However, the trials can be separated into target present trials and target absent trials, with a different network for each type. The network for the "present" response has an OR gate, the network for the "absent" response has an AND gate. The task is represented by one OR network and one AND network. More information on representing tasks with networks is given in a later chapter.

The *duration* of an arc x is a nonnegative random variable, $D(x)$. On a particular trial, each arc is assumed to take on a particular value from its probability distribution. The duration of an arc representing a process is the duration of the process. The duration of an arc representing an instantaneous action, such as a resource becoming available, is zero on every trial.

The *duration of a path* is the sum of the durations of all the arcs on it. A path can consist of a single vertex; in that case, the path duration is 0. Since arc durations are random variables, the duration of a path is a random variable also. To be specific, suppose a vertex u precedes a vertex v on a particular path. The durations of the arcs on this path will vary from trial to trial, so the duration of the path will vary also.

If there is more than one path from u to v, and we are interested in the longest path from u to v, the path with the longest duration may not be the same path on each trial. Despite this complication, we can speak of the duration of the longest path from u to v; it is a random variable whose value on a particular trial is the sum of the arc duration values on that path which happens to be the longest for that trial. (On a given trial, there may be several paths tied as longest or shortest from one vertex to another; this turns out to not affect our conclusions.)

The time elapsing between the occurrence of vertex u and the occurrence of vertex v is denoted $D(u,v)$. If all vertices are AND gates, $D(u,v)$ is the duration of the longest path between vertices u and v. On a particular trial, the longest path from the starting vertex of the network, *o*, to the ending vertex, *r*, is called the *critical path*; in an AND network, the duration of the critical path is the response time for the trial. If all vertices are OR gates, $D(u,v)$ is the duration of the shortest path between vertices u and v. The shortest path from vertex u to vertex v is called a *geodesic*. In an OR network, the duration of the shortest path from the stimulus to the response on a trial is the response time for the trial.

If more than one response is made, there will be a response time for each. If one is interested in a particular response, arcs not preceding that response can be ignored because they have no influence on the time at which that response is made. If more than one stimulus is presented, the response time for the subtask associated with a stimulus is the time elapsing from the onset of the stimulus to the response for that stimulus. When two stimuli are presented, they are typically presented in the same order on every trial, separated by a stimulus onset asynchrony. It is sometimes of interest to know the time at which a particular response is made using the time at which the first stimulus was presented as the reference point.

In the next chapter, we turn to *Task Network Inference*, the construction of a directed acyclic task network from observed effects of factors selectively influencing processes in it.

Acyclic Task Networks in Human Factors

A major use of task networks is in Human Factors. A network is often drawn to represent operations of machines in a workplace and it is natural to extend the network to include the cognitive operations of workers interacting with the machines. Large portions of such cognitive task networks can be constructed by observing workers and reasoning about necessary information processing, a procedure called *cognitive task analysis*. One of the best examples of a successful application is Project Ernestine (Gray, John, & Atwood, 1993). In a now well known story, while new workstations were under development for telephone operators, analysts observed videotapes of operators using the old workstations for various tasks. A critical path network was drawn for each task, using the cognitive task analysis method CPM-GOMS. (CPM stands for Cognition, Perception, Movement, and also for Critical Path Method. GOMS stands for Goals, Operators, Methods and Selection Rules.) Estimates of the durations of component processes such as typing and speaking were obtained from the videotapes and from the human factors literature. For each task, a network was drawn for use of the old workstation and another network was drawn for use of the new workstation. With the networks and estimated durations, the time to complete each task could be predicted for the new workstations.

Specifications indicated that several processes would be faster with the new workstations. Surprisingly, predicted times to complete tasks were longer. In the networks, some of the faster processes were not on the critical path, so their shorter durations did not shorten completion time. However, several other processes were inserted into the critical path, thus increasing completion time. When the new workstations were tested, completion times were indeed longer.

GOMS was developed by Card, Moran and Newell (1993), and CPM-GOMS by John (1990). Cognitive task analysis is discussed in Schweickert, Fisher and Proctor, 2003.

Systems Not Easily Represented in Acyclic Task Networks

Systems that cannot be formulated as acyclic AND or OR task networks usually have one of the following features, (1) the absence of discrete events, (2) the presence of feedback, or (3) the wrong kind of gates. For issue (1), if there are no discrete events, then the system can be represented as a directed acyclic task network only in an unenlightening way, as a single arc directed from the stimulus onset to the response. Systems with no discrete events are plausible, but beyond the scope of this work; a special issue of *Acta Psychologica* (1995) has relevant papers. For issue (2) some forms of feedback cycles can easily be reformulated as part of an acyclic task network. For example, if a process is simply repeated a random number of times, and no output is sent to other processes until the last repetition, this entire action can be represented in a network as a single arc with a random duration. Feedback causes a problem for our analysis when a process producing feedback activates processes following it at the same time as it reactivates itself or earlier processes. The problem is that processes cannot then be readily classified as sequential or concurrent.

For issue (3), production systems (*e.g.*, Anderson & Bower, 1974; Meyer & Kieras, 1997a, b) are important examples of systems which often have the wrong kinds of gates for our decomposition. A particular production system might easily be representable as an AND or OR task network. But in most production systems an action starts when a compound proposition becomes true. A problem arises when its truth value depends on an event such as the presence of a goal instead of the event that a process has finished. For gates not using standard Boolean logical operations, decomposition with selective influence may be difficult. An example of a nonBoolean gate is a gate releasing a process when the total activation into it exceeds a threshold, the activation being a continuous quantity. The difficulty does not arise in representing the task as a network of some kind, perhaps as an OP diagram. The difficulty is that there is little hope of finding factors which selectively influence processes when a gate blends outputs of several processes. The hard problem is finding a robust alternative to selective influence. One of our major points is that data can easily lead to rejection of the assumption that a directed acyclic network exists, in which experimental factors selectively influence processes. The price for a class that can be rejected is an inability to model everything.

Processing Trees

Responses can be classified in various ways, as, say, correct or incorrect, and we turn now from the time required to respond to the type of response that is made. One of the most widely used structures for modeling accuracy is a processing tree; uses range from perception (*e.g.*, Ashby, Prinzmetal, Ivry & Maddox, 1996; Prinzmetal, Ivry, Beck & Shimizu, 2002) to social cognition (*e.g.*, Klauer & Wegener, 1998). Batchelder and Riefer (1999) provide an excellent review.

In a processing tree, when a process finished, it produces an outcome with a certain probability, and the next process is selected depending on which outcome occurred. Some outcomes of processes are responses, and these fall into various classes. A processing tree is used to predict the probabilities of the response classes from the probabilities of the various process outcomes. An important kind of processing tree is a multinomial processing tree, in which the parameters satisfy a certain constraint ensuring that parameter estimates have a simple form; for details see Hu and Batchelder (1994).

A widely used processing tree model was proposed by Jacoby (1991) in his process dissociation procedure, see Figure 2.3. Two groups of subjects studied the same two lists of words. After study, they were presented with test words which were from List 1 or List 2 or neither. Subjects were asked to say for each word whether it was old or new. For the inclusion group, a word was considered old if it was in either List 1 or in List 2. For the exclusion group, a word was considered old only if it was in List 2.

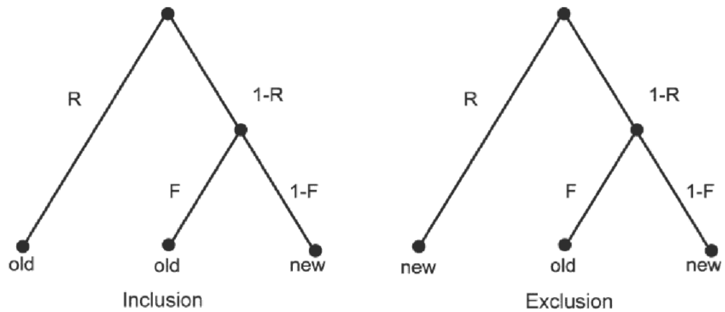


Fig. 2.3. Processing trees for inclusion and exclusion conditions. Arcs are directed from top to bottom.

According to the model, when a subject sees a word at test, he attempts to consciously recollect it. For a word studied on either list, this recollection is successful with probability R, and yields the information that the word was studied and which list it was in. For a studied word, if the word is not recollected, with probability F it is judged familiar.

Consider a word in List 1. A subject in the inclusion group will say the word is old if it is recollected, or if it is not recollected, but judged familiar. That is, for a word in List 1, the subject will say old (correctly) with probability

$$p_{inclusion} = R + (1 - R)F.$$

For a word in List 1, a subject in the exclusion group will not say the word is old if it is recollected. However, a familiar word is more likely to be from list 2 (presented recently), than from list 1 (presented earlier), or new (not presented). So if a word is not recollected, but is judged familiar, the subject will say the word is old. That is, for a word in List 1, the subject will say old (incorrectly) with probability

$$p_{exclusion} = (1 - R)F.$$

The two equations above can be solved for the two unknowns, R and F.

To test the model, experimental factors expected to selectively influence recollection or familiarity are manipulated. For example, Jacoby (1991) proposed that a secondary task carried out during testing would not change the familiarity of items, because familiarity was established during study. However, the secondary task would harm recollection, because recollection occurs during testing. Hence, the secondary task is expected to decrease R leaving F invariant, as was found. An example of a factor that does not selectively influence a parameter is the presentation of words as anagrams instead of in the usual way. This manipulation changes both R and F (Jacoby, 1991; Jacoby, Toth & Yonelinas, 1993).

In a *processing tree*, at each vertex a process is executed. (In a task network, processes were represented by arcs.) The first process to be executed is represented by a special vertex, the *root* (at the top in our illustrations). When a process is executed, it produces one of several possible outcomes. These outcomes are represented by arcs leaving the vertex representing the process. (Because the direction of all arcs is from top to bottom, arrows can be omitted.) Such arcs are called the *children* of the vertex. An arc is directed from its *starting vertex* to its *ending vertex*. Each child of a vertex has a probability associated with it; this is the probability the corresponding output is produced, given that the process represented by the vertex is executed. The sum of the probabilities associated with the children of a vertex is 1. When an output is produced by a process, the arc corresponding to it is said to be *traversed*, the ending vertex of the arc is said to be *reached*, and the process represented by this vertex begins execution. This procedure continues until a vertex with no children is reached. Such a vertex is called a *terminal vertex*, and it produces a response. The responses fall into mutually exclusive classes. Responses made at a particular terminal vertex fall into one such class.

As in a task network, a path from a vertex *u* to a vertex *z* consists of the vertex *u*, followed by an arc directed from *u* to another vertex *v*, followed by an arc directed from *v* to another vertex *w*, and so on, with the last arc having ending vertex *z*. A single vertex is considered a path. A *simple path* is a path in which no vertex is repeated. We say a network is *connected* if for any two vertices *u* and *z* there is a path from *u* to z, or a path *from* z *to* u. A *tree* is network in which for every pair of vertices *u* and *z*, there is exactly one simple path from *u* to z or exactly one simple path from *z* to *u*, but not both. With our definition, a tree is connected. Further, no vertex precedes itself on a path, so a tree is a directed acyclic network. (With a task network, there may be more than one simple path from one vertex to another, but this cannot happen in a tree.)

The probability of a path is the product of the probabilities associated with the arcs on the path. The probability of a path consisting of a single vertex is 1. Given that processing started at the root, the probability a response is made at a particular terminal vertex is the product of the probabilities on the path from the root to that terminal vertex (there is exactly one such path). The probability a response in a particular class is made is the sum of the probabilities that responses are made at terminal vertices associated with that class. For short, we will sometimes say *tree* to refer to a processing tree.

Because multinomial processing trees are so widely used, their statistical analysis is well developed. See, for example, Batchelder and Knapp (2004), Batchelder and Riefer (1986, 1990), Chechile and Meyer

(1976), Hu and Batchelder (1994), and Riefer and Batchelder (1988). Software is well developed also. See, for example, Dodson, Prinzmetal and Shimamura (1998); Hu (1999); Rothkegel (1999); and Stahl and Klauer (2007).

Systems Not Easily Represented As Processing Trees

A tree is a special form of directed acyclic network, so difficulties that arise for the former also arise for the latter. As with directed acyclic task networks, the following are common impediments to forming a processing tree model. (1) Continuous output, rather than discrete output, is not easily represented in trees (Kinchla, 1994; Slotnick, Klein, Dodson & Shimamura, 2000). (2) Many forms of feedback, such as error correcting procedures, cannot readily be represented without cycles. (3) In a tree, the gate for releasing a process is special, because a process has at most a single predecessor. Although this limitation can sometimes be overcome by placing copies of a process at several places in a tree, a factor selectively influencing this copied process may not be well behaved.

Analyzing both reaction time and accuracy

It is natural to attempt to combine a processing tree with a task network, to obtain a model for both reaction time and accuracy. A start has been made for processes in series by Hu (2001) and Schweickert (1985). The difficulty is not so much in finding a common structure, but in deriving predictions for factors selectively influencing processes. A simple example illustrates the problem. Suppose process A requires time $D(A)$ to produce a correct output, and does so with probability $p(A)$. Over all trials, the expected value of the contribution of process A to the reaction time for a correct response is $p(A)E[D(A)]$. A factor selectively influencing A, making it more difficult, has two effects. It will decrease the probability A produces a correct output and it will increase the duration of A. Such opposing effects are hard to work with.

Chapter 3

Selectively Influencing Processes In Task Networks

Although Sternberg (1969) focused on serial processes, he noted that the combined effect of two factors selectively influencing two parallel processes would be the maximum of their individual effects. Effects of factors prolonging processes that are not in series have been studied for a long time (Karlin & Kestenbaum, 1968; Welford, 1952). We know about these effects in more detail now. When factors selectively influence processes in an AND or OR task network, systematic patterns occur in the mean response times. This chapter gives an overview. At the end of the chapter we discuss how a process can be part of a larger superprocess or have constituent subprocesses. For analysis of response times in general, see Van Zandt (2002).

Effects of Selectively Influencing Processes in Task Networks

Figure 3.1 illustrates a model for a dual task in which a subject produces a time interval and, part way through the time interval, searches a screen for a target (Schweickert, Fortin & Sung, 2007). Each trial had two components. In the first component, a tone was presented. The subject encoded its duration, to be used in the second component of the trial as the duration goal of a time interval the subject would produce. When ready for the second component, the subject pressed a button (noted as event o_1 in Figure 3.1). The button press blanked the screen and started the time interval the subject was producing. After an interval (the stimulus onset asynchrony, SOA), a display was presented (noted as event o_2 in Figure 3.1). The subject was to search through the display and decide whether a target (a circle) was present among the distractors (circles with a vertical line stem). The subject was to respond only after he or she believed both that a time interval had elapsed whose duration was the goal duration and also that a decision was made as to whether the target was present in the display or not. The subject made a single response by pressing a button. One button was pressed to indicate that the target was present, another button was pressed to indicate that the target was absent. The model is a simple AND network.

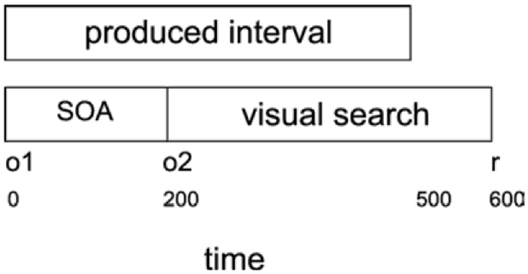


Fig. 3.1. Processes in a dual time production and visual search task. If the produced interval is short, effects of prolonging SOA and visual search will be additive.

Three processes are illustrated, the produced time interval, the SOA, and the visual search. (The visual search could be divided into subprocesses, but the details are not relevant here.) The SOA and the visual search are sequential. The time interval is concurrent with the SOA and concurrent with the visual search.

For simplicity, consider trials on which the target is absent. On such trials, the subject must search all the items in the display to correctly decide the target is absent. Consider effects of manipulating three factors. We can increase the time required for the search by increasing the number of items in the display. We can increase the duration of the SOA directly. Finally, we can increase the duration of the time interval produced by the subject by giving the subject a longer duration goal.

In the initial condition illustrated in Figure 3.1, the goal duration has elapsed (*i.e.*, the produced time interval is over) before the search subtask is completed. The response is made at time 600. If the SOA is increased by 100, the response time is increased by the same amount, 100. If the SOA is returned to its original value and the search is increased by 200, the response time is increased by 200. Finally, if the SOA is increased by 100 and the search is increased by 200, the response time is increased by 300. The combined effect of both factors is the sum of the effects of each of them separately. The factors are additive and one can conclude that there exists a task network in which there is a pair of sequential processes, and each factor selectively influences a different process in the pair.

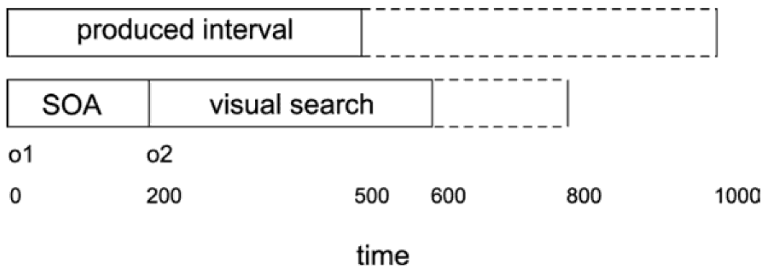


Fig. 3.2. The combined effect of prolonging two concurrent processes will be less than additive.

Figure 3.2 illustrates the effect of selectively influencing both the search and the produced interval. The initial condition is the same as before, with the response made at time 600. As before, when the search is increased by 200, the time at which the response is made increases by the same amount, 200. When the produced interval is increased by 500, the time at which the response is made increases by 400. Finally, when the search is increased by 200 and the produced interval is increased by 500, the time at which the response is made still increases by 400. The combined effect of both factors is smaller than the sum of their separate effects. The two factors interact, and we will see later that from the form of the interaction one can conclude that the task can be represented with an AND network in which there are two concurrent processes, and each of the two factors, produced-interval-goal and display size, selectively influences a different one of the two concurrent processes.

It is straightforward to check that the factors of produced-interval-goal and SOA also interact. The combined effect of both these factors is smaller than the sum of their separate effects. As we will see, from the form of the interaction one can conclude that the task can be represented with an AND network in which the factors of produced-interval-goal and SOA selectively influence two concurrent processes.

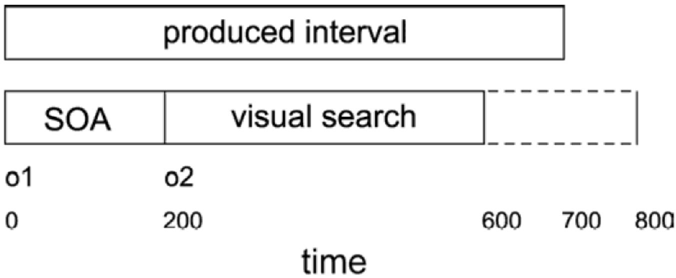


Fig. 3.3. If search is prolonged by 200, reaction time increases by 100. If the produced interval is long, effects of prolonging SOA and visual search will be greater than additive.

With the initial condition in Figure 3.1, SOA and display size have additive effects on reaction time. But with a different initial condition, illustrated in Figure 3.3, these factors could interact. In Figure 3.3 the produced interval is 700, longer than the sum of the SOA and the search duration. With this initial condition, if the SOA duration is increased by 200, the increase in the reaction time is only 100. Likewise, if the search duration is increased by 200, the reaction time increases by 100. Finally, if the SOA duration is increased by 200 and the search duration by 200, the reaction time increases by 300. The combined effect, 300, of prolonging the SOA and the search is greater than the sum of their separate effects (100 + 100). The factors of SOA and display size interact.

In the task network, each of the three factors selectively influences a different process. It is awkward to conclude in this situation that an interaction between two factors indicates that the two factors influence the same process, because the interaction comes and goes depending on the duration of the time interval. However, the interactions are systematic, as we will see when we examine the details of prolonging processes.

Slack

The behavior of AND networks and OR networks is similar, so it will suffice to focus on AND networks. If all the processes were in series, and an amount u were added to the duration of a process A, then the response time would increase by u. But suppose the processes are in an AND network, and the process A is not on the longest path through the network. The response time is determined by processes which bypass A, so incrementing the duration of A by a small amount would have little or no effect on the response time. For example, in the AND network in Figure 3.1, the response time is the duration of the longest path, 200 + 400 = 600. The duration of the produced interval is only 500. If the produced interval is increased by 50, the response time would not change because the longest path did not change. (An analogous situation would arise in an OR network, if a prolonged process is not on the shortest path through the network.)

On a particular trial, if a process A is not on the longest path through an AND network, we say there is slack for the process A on that trial. Suppose we knew the durations of all the processes on that particular trial. And suppose we could rerun the trial with the same process durations, except that the duration of A is prolonged. Then, the longest time by which A could be prolonged without delaying the response r is the *slack* from A to r, sometimes called the *total slack* for A. It is denoted $s(A, r)$. In this notation, the first argument, A, is an arc and the second argument, r, is a vertex, the vertex at which response onset occurs. In Figure 3.1, the total slack for the produced interval is 100.

If all the process durations were known, $s(A, r)$ could be determined, that is, $s(A, r)$ is a function of the process durations. The intuition is that the slack from process A to r is the difference between (1) the duration of the longest path from the start of the network to r and (2) the duration of the longest path that goes from the start of the network to r and that also goes through arc A. Let o denote the starting vertex of the network, and let A' and A'' denote the starting and ending vertices of arc A. For two vertices, say o and A', with the first preceding the second, let $d(o, A')$ denote the duration of the longest path between them. Let $d(A)$ denote the duration of process A. Then the total slack for A is

$$s(A, r) = d(o, r) - d(o, A') - d(A) - d(A'', r).$$

For more detail, see Schweickert (1978). With the formula, one can see that on a particular trial a process is on the longest path from o to r, that is, on the critical path, if and only if its total slack is 0.

Two related quantities are also used in the analysis of sequential processes. Consider an AND network in which process A precedes process B. The largest amount of time by which A can be prolonged without delaying the start of B is the *slack from A to B*. Its value can be found in the following way. Remove from the network all processes that do not precede B', the starting vertex of B. (This includes removing process B itself, but leaving vertex B'.) In the remaining network, B' can be considered the terminal vertex. Then, by analogy with finding the total slack for A, the slack from A to B is

$$s(A, B') = d(o, B') - d(o, A') - d(A) - d(A'', B').$$

Now, restore the removed processes to the network, and suppose A is prolonged by an amount just long enough to make B start late, that is, A is prolonged by exactly $s(A, B')$. How much of the total slack for A remains? This quantity is the *coupled slack* from A to B,

$$\begin{aligned} k(A, B) &= s(A, r) - s(A, B') \\ &= d(o, r) - d(o, B') - d(A'', r) + d(A'', B'). \qquad (3.1) \end{aligned}$$

This quantity can be positive, zero, or perhaps contrary to intuition, negative. We will see that its value determines the form of the interaction between factors selectively influencing processes A and B. Here are examples of coupled slack values in later figures. In Figure 3.4, $s(A, r) = 225$ and $s(A, B) = 440$, so $k(A, B) = 225 - 440 = -215$. In Figure 3.7, the slack from B to C is 125, the same as the total slack for B. Hence, $k(B, C) = 0$. However, if the duration of C were 100 (instead of 375 as in the figure), the total slack for B would be 175. The slack from B to C would still be 125. Hence, the coupled slack for B and C would be $k(B, C) = s(B, r) - s(B, C) = 175 - 125 = 50$.

Typically in an experiment we do not know the durations of individual processes on a trial, so we do not know the value of the slack for any process. Values of this unobservable quantity are assumed to have a probability distribution over all the possible trials. The random variable taking on these values is denoted $S(A, r)$; it is a function of the random variables which are the process durations.

Selective influence

There are many ways that changing the level of an experimental factor might selectively influence the duration of a process. For example, a factor might make the duration of a process more variable, without changing its mean. It is reasonable to assume that if changing the level of a factor makes a process more difficult, it increases the mean duration of the process. Unfortunately, this simple assumption does not lead to many useful conclusions, so stronger assumptions are needed (Townsend, 1990). Different assumptions about selective influence are needed for different purposes. This chapter is concerned with expected values of reaction times, so the assumptions need not be strong. For many conclusions about expected values, dependencies between random variables can be ignored; for example, the expected value of $X + Y$ is the sum of the expected values of X and Y, whether X and Y are correlated or not.

Consider a factor selectively influencing a process A. Let a level of the factor be denoted i, for $i = 1, 2, \ldots$. If the brightness of a stimulus is the experimental factor, then the levels 1 and 2 might indicate bright and dim, respectively. Higher level numbers indicate greater process difficulty (for both AND and OR networks).

When the factor selectively influencing process A is at level 1, in the initial condition, the duration of A is a random variable A_1. (Random variables will usually be denoted by capital letters, values they take on by corresponding small letters.) We assume that an increase to level 2 of the factor adds something to the duration of A (for both AND and OR networks). That is, there is a nonnegative random variable U such that at level 2 of the factor the duration of A is $A_2 = A_1 + U$. (The next chapter supplies more details.) Increasing the level of the factor from 1 to 2 is said to *increment the duration* of A. One immediate consequence is that the expected value of the duration of A at level 2 is greater than or equal to the expected value at level 1.

Sternberg (1966, 1969) gives an example of how this assumption would be met in practice. Suppose search through a memory set is serial and exhaustive, that is, items are processed one by one, and every item is processed. If the memory set is {a, b} in one condition, and {a, b, c} in another, then increasing the size of the memory set increments the duration of the memory search.

This assumption is equivalent to another one (see Müller & Stoyan, 2002; Townsend & Schweickert, 1989). When the factor is at level 1, let the cumulative distribution function of the duration of process A be $F_{A1}(t) = \text{Prob}[A_1 \leq t]$. Likewise, let $F_{A2}(t)$ be the cumulative distribution function of the duration of process A when the factor is at level 2. Then increasing the level of the factor from 1 to 2 increments the duration of A if and only if $F_{A1}(t) \geq F_{A2}(t)$.

If at every t, the cumulative distribution function for one random variable is greater than or equal to the cumulative distribution of another, then the former is said to be *stochastically smaller than* the latter. (Note that the larger cumulative distribution function produces the smaller mean.) When we say a factor selectively influences a process A, one assumption we make is that when the level of the factor is increased from i to i', the duration of A at level i is stochastically smaller than the duration of A at level i'.

When we say each of two factors selectively influences a different process, one assumption we make is that each factor increments the duration of a different process; that is, the marginal cumulative distribution functions of the two process durations are each ordered by the levels of the factors. What about the joint distribution of the process durations? An easy assumption to make, but a strong one, is that the durations of all the processes are mutually stochastically independent at every combination of factor levels (see, *e.g.*, Schweickert & Giorgini, 1999; Schweickert, Giorgini & Dzhafarov, 2000). Weaker assumptions about selective influence sufficient for the results presented in this chapter are given in the next chapter. It formulates in a more precise way assumptions originally given in Schweickert (1982), Schweickert and Townsend (1989), Townsend and Schweickert (1989), and Schweickert and Wang (1993). Recently, general formulations of selective influence have been developed by Dzhafarov (1996) and Kujala and Dzhafarov (2008). These will be discussed in a later chapter.

The next chapter deals with the following difficulty. Suppose a subject is presented with a block of trials with a factor at level 1, and later is presented with a block of trials with the factor at level 2. It is no problem to subtract the mean reaction time at level 1 from the mean reaction time at level 2. But a problem arises if we consider subtracting individual reaction times at level 1 from individual reaction times at level 2. For a given trial with factor level 2, which trial with factor level 1 do we subtract from it? We do not have a sample of pairs of reaction times, with the only difference between one element of a pair and the other being a change in the duration of process A. In particular, it is impossible in the experiment to obtain a sample $<a_1, a_2>$ of an observation a_1 of A_1 paired with an observation a_2 of A_2. It turns out that the assumption that each factor increments the duration of a different process can be formulated in such a way as to imply the existence of a common theoretical probability space for the random process durations at all levels of the factors, whether or not we can make experimental observations at all levels simultaneously. Details are in the next chapter.

Monotonic Response Time Means

We are now in a position to explain what happens to the response times when a factor selectively influences a process by incrementing its duration. Consider an AND network with response made at r, and consider a particular trial with the factor selectively influencing process A at level 2. When a trial occurs, a sample value is taken from the population distribution of each random variable's process duration. On this particular trial, then, every process has a duration which is a nonnegative number. In particular, the duration of A is $d(A) + u$, for some value $d(A)$ of the duration of A when the factor is at level 1 and some value u of the duration of the increment. The durations of the remaining processes $C1, ..., Cp$ are $d(C1), ..., d(Cp)$. The duration values can be used to calculate the values of quantities not only for the trial at level 2 of the factor, but for what would have happened if the trial had been at level 1 of the factor.

On a particular trial, the slack from A to r at level 1 of the factor has a particular numerical value, $s(A, r)$. If the increment u is less than the slack from A to r, there is no increase in the response time produced by changing the factor from level 1 to level 2. If u is greater than the slack from A to r, a portion of u would be used to overcome the slack from A to r, and what remains of u would increase the response time. That is, the increase in the response time would be

$$\begin{array}{ll} 0 & \text{if } u \leq s(A, r) \\ u - s(A, r) & \text{if } u > s(A, r). \end{array}$$

It is convenient to use the notation

$$[x]^+ = 0 \text{ if } x \leq 0$$
$$[x]^+ = x \text{ if } x > 0.$$

With this notation, the increase in response time when A is prolonged by u, is $[u - s(A, r)]^+$.

The process durations vary from trial to trial, so they are random variables. In the initial condition, the slack from A to r is a function of the random variable process durations, so it too is a random variable, $S(A, r)$. In the expression, we use the capital letter S to denote slack as a random variable, and a small letter s to denote a numerical value the random variable takes on. When the factor selectively influencing A is at level 2, over all the trials the amount by which A is prolonged beyond its duration at level 1 is a nonnegative random variable, U. (We are assuming the factor selectively influencing A increments its duration.) Over all the trials, the expected value of the response time, E[T], is increased by a nonnegative amount $E([U - S(A, r)]^+)$. The result is that as the factor levels increase, the mean response times increase monotonically.

To give more detail, if we let T_1 and T_2 be the response times when the factor influencing A is at levels 1 and 2, respectively, then

$$E[T_1] \leq E[T_1] + E([U - S(A, r)]^+) = E[T_2]. \tag{3.2}$$

The result is that when the process A is prolonged, its mean response time either increases or stays the same; *i.e.*, it increases monotonically. The reasoning is similar for other changes in the factor levels.

A note on SOA in dual tasks

When there are two responses, it is customary to use the time from the onset of the second stimulus to the onset of its response as the reaction time to the second stimulus. Then for the model in Figure 2.1 it is not hard to show that the response time to the second stimulus decreases monotonically as the SOA increases. This may seem at first to contradict the statement that increasing a factor level increases the mean response time. However, with this way of measuring the response time to the second stimulus, the location of the event used to start the clock (the onset of stimulus 2) changes as the SOA changes. If instead the clock is started at the onset of stimulus 1, the mean time at which the response to the second stimulus is made increases as the SOA increases.

A note on OR networks

In an OR network with response made at r the greatest amount of time by which a process A may be shortened without decreasing the response time is called the *surplus* from A to r, analogous to the slack from A to r.

In an OR network, the mean response times decrease monotonically as process durations decrease (Schweickert & Wang, 1993). This is equivalent to saying that mean response times increase monotonically as process durations increase, *i.e.*, as the factor levels increase, the same result as for AND networks. Because results about shortening can be rephrased as results about prolonging, we speak of factors prolonging process durations, for both OR and AND networks.

Monotonic Interaction Contrasts

Consider a factor $\mathcal{A}$ selectively influencing a process A and another factor $\mathcal{B}$ selectively influencing a different process, B. Let the levels of the factor selectively influencing process A be denoted $i = 1, 2, \ldots$, and let the levels of the factor selectively influencing process B be denoted $j = 1, 2, \ldots$. In both cases, higher numbers indicate greater process durations (for both AND and OR networks). When the first factor is at level i and the second at level j, we denote the response time as T_{ij}, with expected value $E[T_{ij}]$. For each combination of levels (i, j) we define an interaction contrast

$$(\mathcal{AB})_{ij} = E[T_{ij}] - E[T_{1j}] - E[T_{i1}] + E[T_{11}]. \tag{3.3}$$

When the processes are in series, the factors have additive effects, so the interaction contrasts are zero for every i and j.

The effects of selectively influencing two processes in a task network depend on how the two processes are arranged. The major distinction is between concurrent and sequential pairs of processes. (For a good introduction, see Logan, 2002.) Sequential pairs are further distinguished depending on whether they are in a structure called a Wheatstone bridge. A Wheatstone bridge is illustrated in Figure 3.4. Processes A and B are on opposite sides of the bridge. One place a Wheatstone bridge arises is in a dual task, when subjects are instructed to respond to the first stimulus before responding to the second stimulus. This commonly given instruction, in effect, inserts a dummy process between the two responses to establish their order. Figure 3.5 shows the task network for the dual task model in Figure 2.2 drawn with the additional constraint that response 1 precedes response 2. Figure 3.5 can easily be redrawn as an AND network. It has the form of a Wheatstone bridge. For analyzing times to make response 2, processes $B1$ and $B2$ are on opposite sides of a Wheatstone bridge (as are other pairs such as $A1$ and $C2$). Other examples of models with a Wheatstone bridge are the double bottleneck models of de Jong (1993), Ehrenstein, Schweickert, Choi and Proctor (1997) and the stimulus-response compatibility model of Kornblum, Hasbroucq, & Osman, (1990).

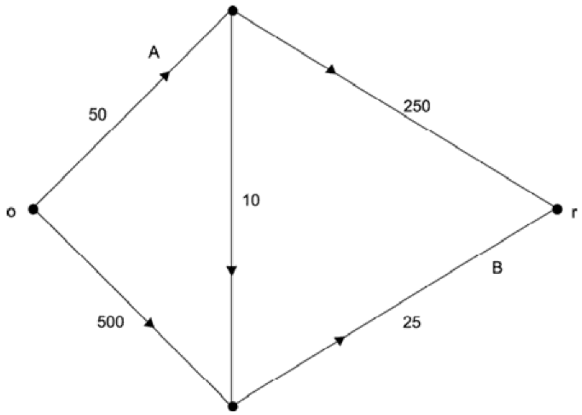


Fig. 3.4. Processes A and B are on opposite sides of an incomplete Wheatstone bridge.

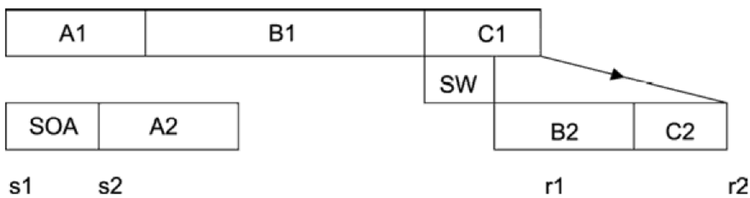


Fig. 3.5. Instructing the subject to make response $r1$ before $r2$ creates a Wheatstone bridge.

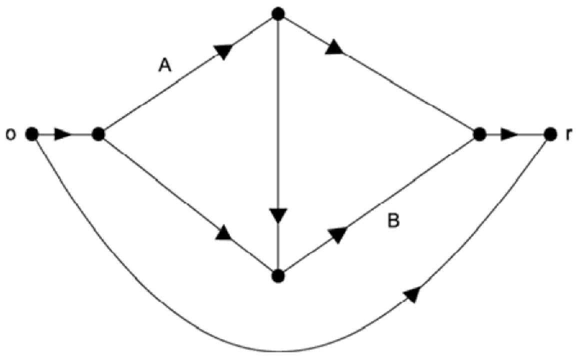


Fig. 3.6. Processes *A* and *B* are on opposite sides of a complete Wheatstone bridge.

Pairs of sequential processes are subdivided into those not on opposite sides of a Wheatstone bridge, those on opposite sides of an incomplete Wheatstone bridge (Figure 3.4), and those on opposite sides of a complete Wheatstone bridge (Figure 3.6).

We will first discuss interactions indicating concurrent processes and then discuss those indicating sequential processes. Before discussing interactions, we explain our simulations.

Calculations and simulations

A number of practical questions arise when one considers testing these predictions in experiments. Are the effects big enough to be found? Will a reasonable number of trials be sufficient for discerning the patterns? To investigate the feasibility of finding these patterns in data, we produced results for hypothetical experiments, by simulation and by calculation. These examples refute objections that the interactions predicted by the theory are small and easily mistaken for additivity (Molenaar & van der Molen, 1986; Vorberg & Schwarz, 1988).

The predictions about means and interaction contrasts are distribution free. But are the predicted patterns more conspicuous for some distributions than others? To investigate this possibility, we used two different distributions for process durations, the exponential and the truncated normal. The first is highly skewed while the second is nearly symmetrical. Little is known about the actual distributions of individual mental processes, but normal and exponential distributions are plausible and often assumed. A normal distribution would be expected if the duration of a mental process were the sum of many components. There is evidence in some experiments for exponential distributions (or sums of these), *e.g.*, Ashby and Townsend (1980) and Kohfeld, Santee, and Wallace (1981), although Sternberg (1964) found evidence against them. For more discussion of distributions, see Luce (1986).

In examples using exponential process durations, the expected values of the response times were calculated exactly with the OP diagrams described in a later chapter (Fisher, 1985; Fisher & Goldstein, 1983; Goldstein & Fisher, 1991, 1992). For examples using truncated normal distributions, no algorithm giving exact values of expected values is known, and the results are based on simulations using MICROSAINT (Micro Analysis and Design, 1985).

For each type of distribution in our hypothetical experiments, the process durations were assumed to be mutually independent, that is, the joint distribution for every subset of processes was assumed to be the product of the corresponding marginal distributions. Independence is not a realistic assumption, and the predictions do not require it. Little is known about the actual correlations between durations of mental processes, so the choice of correlation values is somewhat arbitrarily. We chose 0 (independence) because it is familiar and intuitively clear. Later we will relax this assumption.

Interaction Contrasts: Concurrent Processes

When two factors selectively influence concurrent processes in an AND network, the following results are predicted: (1) mean response times will increase monotonically with increases in levels of the factors; (2) interaction contrasts will all be less than or equal to zero; and (3) interaction contrasts will decrease monotonically as the levels of either of the factors is increased. Prediction (3) is a consequence of (2). All interaction contrasts calculated for higher factor levels with respect to lower factor levels are predicted to be nonpositive. These predictions are derived in the next chapter.

Example 1: Exponential distributions

If two factors prolong different concurrent processes *A* and *B*, the pattern of interactions produced on response times can be easily seen. Consider the acyclic task network in Figure 3.7. Each process duration was assumed to have an exponential distribution and the durations were assumed to be mutually independent. The processes prolonged, *A* and *B*, are concurrent. Means for *A* and *B* are given in Table 3.1, means for the other processes are as indicated in Figure 3.7. Expected values of response times are in Table 3.1. They were computed from the associated OP diagrams using the algorithm we describe in a later chapter. Note that these numbers are not the means of simulated trials; the algorithm calculates the exact expected values. The interaction contrasts defined in Equation 3.3 are easily calculated; for example, for the change from level 1 to level 2 of each factor, $(\mathcal{AB})_{22}$ = 733.340 – 642.778 – 710.901 + 616.093 = – 4.246. These values are also in Table 3.1. (Note that T_{11} = 616.092, T_{22} = 733.340, T_{12} = 642.778 and T_{21} = 710.901).

The three patterns are immediately apparent. (1) Means are monotonically increasing from left to right and from top to bottom, (2) interaction contrasts are all negative, and (3) they too are monotonic. (All interaction contrasts calculated for higher factor levels with respect to lower factor levels are predicted to be nonpositive, not only those in the table of interaction contrasts.)

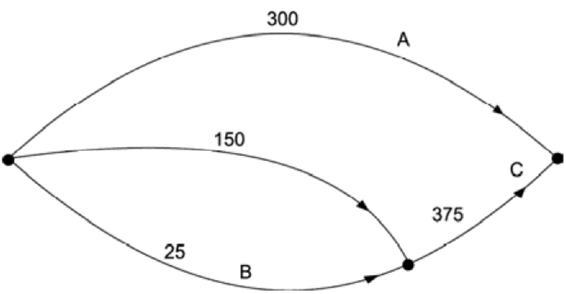


Fig. 3.7. AND network used in simulations. Mean durations of processes not prolonged are on arcs; mean durations of processes prolonged are in table headings.

Table 3.1
Expected Values of Reaction Times
When Factors Influence Concurrent Processes *A* and *B* in Figure 3.7
All Process Durations Exponentially Distributed

μ_A \ μ_B	25	100	150	200	250
300	616.1	642.8	671.1	704.5	741.3
450	710.9	733.3	757.8	787.1	820.0
500	746.5	767.8	791.1	819.2	850.9
550	783.6	803.8	826.1	853.1	883.6
650	861.4	879.8	900.3	925.2	953.6
700	901.8	919.4	939.0	963.1	990.5
750	943.0	959.9	978.8	1002.0	1028.5

Interaction Contrasts

μ_A \ μ_B	25	100	150	200	250
300	-	-	-	-	-
450	-	-4.2	-8.1	-12.2	-16.1
500	-	-5.4	-10.4	-15.6	-20.8
550	-	-6.4	-12.5	-18.8	-25.1
650	-	-8.3	-16.1	-24.5	-32.9
700	-	-9.1	-17.7	-27.1	-36.4
750	-	-9.8	-19.2	-29.4	-39.7

Example 2: Truncated normal distributions

The same patterns were found with each process duration sampled from a truncated normal distribution, that is a distribution whose density function is the normal distribution restricted to nonnegative values, and renormalized so the area under it is one. The standard deviation of each process duration was set to one fourth of its mean. The value of one-fourth is representative of values typically found for response times themselves (see Luce, 1986, p. 64), so we used it for the individual process durations. When the means for processes *A* and *B* were increased, the standard deviations were also increased to one fourth of the new mean to simulate the finding that response time variability typically increases as the mean increases. The process durations were mutually independent.

Simulated response times are given in Table 3.2 for the same network (Figure 3.7) used in the preceding tables. Two thousand simulated trials were run for each combination of means for *A* and *B* using the MICROSAINT system for personal computers (Micro Analysis and Design, 1985). The means and standard deviations (prior to truncation) are the row and column labels in Table 3.2. The interaction contrasts are in the body of the table.

The same three patterns occur as before, although not without exception. The means increase monotonically from left to right and from top to bottom (for the most part), the interaction contrasts are negative (all), and they too are monotonic (for the most part). As noted, some small exceptions occur for the response times and interactions in the first few rows and columns. These arise from sampling the reaction times. It is clear that increasing the mean for *B* from 25 to 100 had little effect on the reaction times, because the increase is not enough to overcome the total slack for *B*. These exceptions would not occur in the population values, although, of course, the effects would still be small.

One of our assumptions in deriving the three patterns is that each factor selectively influences a process by incrementing its duration. The reader may wonder if this form of selective influence occurs here, where a factor increasing the mean duration of a process also increased the variance. It is easy to verify that if two normally distributed random variables have respective means μ_1 and μ_2 and standard deviations σ_1 and σ_2, their cumulative distribution functions cross at $t = (\mu_1\sigma_2 - \mu_2\sigma_1)/(\sigma_2 - \sigma_1)$. Here, since each standard deviation equals the same fraction of the corresponding mean (one-fourth, in this case), the value of t is 0. The distributions were truncated at 0 to avoid negative durations, and since the cumulative distribution functions do not cross elsewhere, they are always ordered in the same way; that is, selective influence takes place by incrementing the process duration (Townsend & Schweickert, 1989).

Table 3.2
Means of Simulated Reaction Times
When Factors Influence Concurrent Processes *A* and *B* in Figure 3.7
All Process Durations Have Truncated Normal Distributions

μ_A	σ_A \ μ_B / σ_B	25 6.25	100 25.0	150 37.5	200 50.0	250 62.5
300	75.0	528	531	547	584	628
450	112.5	556	554	568	596	636
500	125.0	576	577	589	613	648
550	137.5	607	606	616	635	664
650	162.5	679	679	685	696	716
700	175.0	720	720	724	733	750
750	187.5	767	766	767	776	788

Interaction Contrasts

μ_A	σ_A \ μ_B / σ_B	25 6.25	100 25.0	150 37.5	200 50.0	250 62.5
300	75.0	-	-	-	-	-
450	112.5	-	-5	-7	-16	-20
500	125.0	-	-1	-5	-19	-28
550	137.5	-	-3	-10	-29	-43
650	162.5	-	-3	-13	-40	-62
700	175.0	-	-3	-15	-43	-70
750	187.5	-	-4	-19	-47	-78

OR networks

The same patterns are predicted for prolonging concurrent processes in an OR network, except that the interaction contrasts are nonnegative (Schweickert & Wang, 1993).

Statistical considerations

A table whose rows and columns are monotonically increasing is said to satisfy *independence*. This property is of interest in conjoint measurement, so it has been studied in some detail. Although it may seem at first to be a weak condition, independence is quite constraining. Suppose the cells in a table with r rows and c columns are rank ordered. A formula for the number of such tables satisfying independence was derived by Arbuckle and Larimer (1976); they note that the proportion of tables satisfying independence is quite small, even for a small number of rows and columns. Of course, one can always permute the rows and columns until the cell means in the first row are monotonically increasing, as well as those in the first column. McClelland (1977) calculates that there are 3.33×10^6 tables with 3 rows and 4 columns in which the first row and first column are monotonically increasing. Of these, only 462 have the remaining rows and columns monotonically increasing. Independence is unlikely to occur by chance.

To reject independence, it is sufficient to reject the hypothesis that some particular pair of cells is in the proper order. If a given pair was of interest for some reason before the experiment was done, the hypothesis could be tested with a simple a priori test of a contrast. If an out of order pair was located when examining the data, the hypothesis that the population means for those cells are out of order could be tested with an aposteriori test (Kirk, 1982); the appropriate type of aposteriori test would depend on the circumstances.

Interaction contrasts: Sequential processes

When two factors selectively influence two sequential processes, the interaction contrasts defined in Equation 3.3 display simple patterns analogous to those for concurrent processes. Once again, the only difference between AND networks and OR networks is in the signs of the interaction contrasts, as explained below. Details depend on the way the sequential processes are arranged in the network, and are best explained by examples. There are three cases to consider, depending on whether or not the processes A and B are arranged in a Wheatstone bridge. This structure is illustrated in Figures 3.4 and 3.6 in the incomplete and complete form, respectively. More information about sequential processes is in the next chapter.

Sequential processes case 1: Not in a Wheatstone bridge

We begin with the simplest case, sequential processes not on opposite sides of a Wheatstone bridge. In Figure 3.7, processes B and C are an example.

Example 3: Exponential distributions

Table 3.3 gives mean response times and interaction contrasts for an AND network when processes B and C were prolonged. All process durations were assumed to be exponentially distributed and mutually independent. The mean for process A was 300, the means used for B and C are in the table.

Three patterns for interaction contrasts are apparent in the table: (1) mean response times are monotonically increasing across the rows and down the columns, (2) interaction contrasts are all positive (or zero), and (3) interaction contrasts are monotonically increasing across the rows and down the columns. We do not show all possible interactions contrasts, but all calculated for higher factor levels with respect to lower factor levels are predicted to be positive or zero, and this implies result (3). If all the gates were OR gates, corresponding patterns are predicted, the difference being that the interaction contrasts would all be negative or zero, so the interaction contrasts would be monotonically decreasing across the rows and down the columns (Schweickert & Wang, 1993).

Example 4: Truncated normal distributions

The same patterns would be found for any other joint density for the process durations and prolongations when the factors selectively influence processes that are sequential, but not on opposite sides of a Wheatstone bridge. For example, Table 3.4 gives the results of simulations in which the same two processes as before are prolonged, but the durations of all processes in the network have mutually independent truncated normal distributions. The simulations were carried out in

MICROSAINT (Micro Analysis and Design, 1985). The same three patterns are evident in the tables. The small negative interaction contrasts in Table 3.4 are based on sample means, and would not occur with population means.

Table 3.3
Expected Values of Reaction Times
When Factors Influence Sequential Processes B and C in Figure 3.7
All Process Durations Have Exponential Distributions

μ_C \ μ_B	25	100	150	200	250
100	376.5	421.3	456.7	495.0	535.5
150	397.7	445.0	481.7	521.0	562.3
200	424.7	474.3	512.1	552.3	594.3
250	455.9	507.5	546.3	587.3	629.9
300	527.3	582.2	622.5	664.7	708.4
375	546.5	602.1	642.7	685.2	729.1
400	566.1	622.3	663.3	706.1	750.1

Interaction Contrasts

μ_C \ μ_B	25	100	150	200	250
100	-	-	-	-	-
150	-	2.6	3.8	4.8	5.7
200	-	4.9	7.3	9.2	10.7
250	-	6.9	10.2	12.9	15.1
300	-	10.2	15.1	19.0	22.2
375	-	10.9	16.1	20.3	23.7
400	-	11.5	17.0	21.5	25.1

Table 3.4
Means of Simulated Reaction Times
When Factors Influence Sequential Processes B and C in Figure 3.7
All Process Durations Have Truncated Normal Distributions

μ_C	σ_C \ μ_B / σ_B	25 / 6.25	100 / 25.0	150 / 37.5	200 / 50.0	250 / 62.5	500 / 125.0	600 / 150.0
100	25.0	314	316	322	339	371	602	702
150	37.5	337	336	347	372	410	651	752
200	50.0	367	370	383	414	458	702	798
250	62.5	410	411	427	459	503	747	851
300	75.0	454	456	472	509	552	803	901
375	93.8	527	530	548	584	627	873	972
400	100.0	550	553	575	608	652	899	997

Interaction Contrasts

μ_C	σ_C \ μ_B / σ_B	25 / 6.25	100 / 25.0	150 / 37.5	200 / 50.0	250 / 62.5	500 / 125.0	600 / 150.0
100	25.0	-	-	-	-	-	-	-
150	37.5	-	-3	2	10	16	26	27
200	50.0	-	0	8	21	33	47	42
250	62.5	-	-1	9	25	36	50	53
300	75.0	-	0	11	31	41	62	59
375	93.8	-	0	13	32	43	58	58
400	100.0	-	1	17	33	45	61	59

Monotonicity of the response times with the factor levels was discussed above. Schweickert and Townsend (1989, Theorem 3) showed that when factors $\mathcal{A}$ and $\mathcal{B}$ selectively influence sequential processes *A* and *B* not in a Wheatstone bridge, the expected interaction contrast $(\mathcal{AB})_{ij}$ is typically positive and always nonnegative. If all the gates were OR gates, $(\mathcal{AB})_{ij}$ is typically negative and always nonpositive (Schweickert & Wang, 1993). It follows that the expected interaction contrasts will be monotonic with the factor levels.

In the AND network examples just given, two factors prolonging sequential processes produce positive interactions. (By a positive interaction, we mean the combined effect of both factors is greater than the sum of their individual effects.) Since factors prolonging concurrent processes produce negative interactions, it might seem that the sign of the interaction is diagnostic for concurrent and sequential processes. However, the situation is more complicated, because factors prolonging sequential processes in an AND network can also produce negative interactions. This is possible only when the two sequential processes are on opposite sides of a Wheatstone bridge (Schweickert 1978), which we now turn to.

Sequential processes case 2: An incomplete Wheatstone bridge

The task network illustrated in Figure 3.4 has an unusual feature. There are three paths through the network, and only one of them contains both A and B. If the arc from A to B has a short duration, then the path containing them both will hardly ever be the critical path, so it will appear as if A and B are not on a path together. In other words, although A and B are in fact sequential, they might appear to be concurrent.

When factors selectively influence sequential processes on opposite sides of an incomplete Wheatstone bridge (*e.g.*, A and B in Figure 3.4), the resulting patterns of mean response times can be similar to (or identical to) the patterns observed when concurrent processes are influenced. Fortunately, the patterns will be different provided a wide range of levels of the factors are used, when large increments in process durations overcome the relevant slacks. Once again, the patterns to be expected are best illustrated by examples, which we will turn to after explaining more about Wheatstone bridges.

The only way two factors selectively influencing sequential processes A and B in a directed acyclic task network can produce a negative interaction is for the network to contain a subnetwork in the shape of a Wheatstone bridge, with A and B on opposite sides of the bridge (Schweickert, 1978). To be more precise about what it means for one graph to have the same shape as another, we need to explain what is meant by two graphs to be homeomorphic. Consider a graph consisting of two arcs in series, one from a vertex u to a vertex v, and another from