3. INDUSTRIAL/HUMAN FACTORS

Chaired by David Washburn, *Georgia State University*

A fuzzy system for mapping worker attributes to tasks

J. PHILIP CRAIGER and MICHAEL D. COOVERT
University of South Florida, Tampa, Florida

We developed a prototype of a rule-based expert system that facilitates a mapping from tasks comprising a job to the knowledge, skills, and abilities (KSAs) ostensibly required to perform the tasks. A variant of fuzzy logic was employed in the inference process. Signal detection theory was used to evaluate the performance of the system. The results indicated that the system performed well in terms of its ability to discriminate between the KSAs that are required of a task and those that are not. The discussion focuses on implications of the system and its use as an intelligent assistant in human resources planning.

Matching tasks to the worker attributes (i.e., knowledge, skills, and abilities, or KSAs) required to perform them is an integral part of every job analysis. This information is required for many aspects of human resources planning, including selection, placement, and classification decisions, and in training and job design (Levine, 1983). The most common method of mapping from tasks to worker requirements involves subject-matter experts who provide judgments of the KSAs that are required to perform tasks. This process is commonly recognized, however, as being problematic. Typically, there is only moderate agreement between the experts' judgments, and the unreliability in the ratings of individual judges (e.g., in how they rate criticality, importance, and frequency) often requires mean ratings to be calculated from large groups of judges, to average it out (Fleishman & Mumford, 1991).

Rules for Mapping KSAs to Tasks

Our research involves a computerized task-to-KSA mapping implementation—that is, a computer program containing structured knowledge about relationships between tasks and associated KSAs (Craiger, 1992). This system is based on the premise that predictable relationships exist between the terminology employed to describe job behavior and the worker attributes required to perform the tasks. This system is a prototype of a larger system that can be employed in concert with, and perhaps in place of, the traditional mapping process outlined above.

Knowledge of the relationships between tasks and worker requirements is implemented via production rules. Production rules are a form of knowledge representation commonly employed in artificial intelligence—a subfield of computer science devoted to the study of intelligent computer behavior (Winston, 1984). They are the most common form of knowledge representation employed in expert systems—computer programs with the ability to emulate expert decision making.

Production rules are two-part structures, consisting of an IF (antecedent) clause and a THEN (consequent) clause. If the antecedent conditions of a rule are satisfied, the rule fires, and its consequent clause is instantiated. Rules are a flexible and natural means of representing knowledge, from the simplest everyday problem to the most complex problem domains (cf. MYCIN, an expert system used to diagnose bacterial infections of the bloodstream; Buchanan & Shortliffe, 1984).

Below, we discuss two potential bottlenecks in the development of our task-to-KSA mapping system. The first involves the problem of providing a computer program with the ability to “derive” meaning from natural language input. The second involves what we call the strict versus heuristic mapping problem.

Natural Language Processing

Before an input represented in natural language form can be processed by a computer there must exist some formal mechanism for semantic processing. To process meaning often requires much in the way of background information, commonsense reasoning, and inference. We argue that our system need not “understand” subtle semantic differences across task statements; rather, the

Correspondence should be addressed to J. P. Craiger, Organizational Systems Department, Navy Personnel Research and Development Center, 53335 Ryne Road, San Diego, CA 92152-7250.

major requirement is that the program should be able to identify the components from the task statement that are essential to the description of job behavior. We define this universe of terminology as the *task lexicon*—the idiomatic terminology employed in describing job behavior.

A lexicon is essentially a dictionary. Just as every language has its own dictionary, so every job has its own, often distinct, lexicon. We consider this terminology idiomatic because the meaning of a particular descriptive term may differ with the job; meaning is context dependent. For instance, the term “assists” may denote (or connote) several distinct types of job behavior, depending upon whether the worker is a dental hygienist, a baker’s assistant, a nurse, or an astronaut. Intuitively, however, we might suspect that the worker attributes required for a task that involves “assisting” will be more similar for the hygienist and nurse than for the other jobs, although the specific job behavior may not be the same. This does suggest, however, that we can quantify the similarity between tasks based on the similarity between the worker requirements.

If we parse a task statement, considering each element (part of speech) independently, we find that certain elements convey more meaningful information about job behavior than others. The goal is to identify elements from the task lexicon that are essential in the description of job behavior. According to guidelines defined by the Functional Job Analysis (Fine & Wiley, 1971) and the Work Performance Survey System (Gael, 1983), three elements to be defined explicitly in every task statement are (1) the main verb that represents the action of the worker, (2) the object (noun) of the verb, and, if required, (3) the work aids or tools used in the task. In our research, these elements will be referred to as the *action verb*, the *object noun*, and the *work aid*, respectively, and we call this combination a *task-statement kernel*, or TSK for short. For example, a production rule in TSK form would be:

IF TSK is: [types forms word processor]
 THEN the following KSAs are required:
 Knowledge of word processors
 Knowledge of spelling
 Typing skills

The antecedent clause contains a TSK and the consequent clause contains the worker attributes required for performance of the task. This example illustrates how production rules facilitate a mapping from a task statement to the worker requirements.

Strict Versus Heuristic Mappings

The second bottleneck concerns the *strict versus heuristic mapping problem*. In the current context, a strict mapping requires that a TSK must match all three elements from the antecedent clause of a rule: If the TSK for a task statement is [types forms word processor], then a rule in the rule database must also contain the antecedent clause [types forms word processor] for a match to exist. A dilemma occurs, however, when there is no rule whose antecedent clause matches all three TSK elements. What if only two TSK elements match, or perhaps only one?

An alternative to strict mapping is heuristic mapping. A heuristic mapping requires a mechanism that allows us to infer or deduce the worker attributes associated with the task. This deductive process is performed by finding the rule with the antecedent clause that *best* matches the task statement under consideration. We employ a variant of fuzzy set theory as the basis of the inference process.

Fuzzy Set Theory

Fuzzy set theory was developed to address problems involving vague, ambiguous, and imprecise information (Kosko, 1992; Zadeh, 1965). Unlike classical set theory, where an element either is or is not a member of a set, fuzzy set theory allows grades of set membership based over the continuum of real numbers [0, 1]. Table 1 illustrates the fuzzy variable height, represented as the fuzzy sets tall, moderate, and short (Craiger, 1992). Grades of membership—the values in the table’s cells—indicate the extent to which an element represents a particular fuzzy set. The element 7’ has a membership grade of 1.0: It is maximally representative of the fuzzy set tall, weakly representative of the fuzzy set moderate, and unrepresentative of the fuzzy set short. The element 6’ is more evenly distributed, and the grades of membership for the element 5’ indicate that it most likely belongs to the fuzzy set short.

We can apply the concept of fuzziness (i.e., gradations of relationships) to the problem of quantifying task similarity. Job analysts often gather information so that jobs can be classified into job families based on indices of similarity. If we can cluster jobs by similarity, we should also be able to quantify the similarity between tasks. We can imagine that this index may be defined over the real continuum [0, 1], where 0 indicates that worker attributes required of the tasks are totally dissimilar, whereas 1 indicates that tasks require exactly the same worker attributes. We would expect that most indices of similarity would fall between these two values, indicating that there is *some* degree of similarity. Fuzzy set theory facilitates the quantification of the semantic space of task statements, allowing us to place task statements on a continuum of semantic similarity. The notion of the fuzzy relation is critical to our conceptualization of the task-similarity space and computerized mapping implementation. The concept of functional similarity provides a means of expressing a fuzzy relation between two tasks.

Functional Similarity

We define functional similarity as the extent to which two tasks require the same worker attributes. We operationalize functional similarity between two tasks in terms of the functional similarity between the TSK elements—

Table 1
 Membership Grades for the Fuzzy Variable, A Person’s Height

Height	Tall	Moderate	Short
7’	1.0	.1	0
6’	.4	.4	.2
5’	.1	.3	.9

Figure 1. Venn diagram of KSA overlap for TSK elements.

Figure 2. Venn diagram of KSA sets for three disparate job families.

that is, the similarity between the action verbs, between the object nouns, and between the work aids. Figure 1 represents the functional similarity described in terms of the similarity between TSK elements (Craiger, 1992).

U is the set of all KSAs extant. Each paraboloid represents a set of KSAs that is required to perform tasks containing the particular TSK element. For example, the set labeled *computers* represents the set of KSAs required to perform tasks that require a computer as a work aid. The intersections between two TSK elements represent the set of KSAs required to perform tasks that contain these two elements, and the triple intersection $types \cap forms \cap computers$ is the set of KSAs required to perform the set of tasks that contains all three TSK elements.

Figure 2 depicts a more inclusive example of TSK elements (Craiger, 1992). The sets tend to form three distinct clusters, which would be interpreted as three distinct job families: medical/surgical, administrative/secretarial, and heavy-equipment operator. The fact that the three clusters overlap minimally indicates that the KSAs required to per-

form one job are substantially different from those required for a job in a different job family.

The variety of set sizes demonstrates that, relative to other TSK elements, some elements are related to a larger number of KSAs. There are two reasons why the KSA set is larger for certain elements than it is for others: (1) the particular TSK element can be used as more than one part of speech, or job descriptor; and (2) the element is applicable across multiple, disparate job families.

Considering the first explanation, the element *files* can be used as an action verb ("*files reports*"), an object noun ("*checks files for errors*"), or a work aid ("*checks client's creditworthiness using the client's credit files*"). We should expect a positive correlation between the semantic ambiguity of an element (i.e., the number of ways in which the element can be used as a part of speech) and the relative number of KSAs associated with the element.

Considering the second explanation, note that *operates* and *supervises* are the only sets that share a commonality across the three job families. The fact that these sets overlap the three job families indicates that there are tasks in each of the families that contain these action verbs. For instance, a task might be "*operates fork-lift*" for the heavy-equipment operator, "*operates on patients*" for the medical/surgical family, or "*operates a computer*" for the administrative/secretarial family. Thus, *operates* and *supervises* should be interpreted as relatively general in scope (i.e., in their potential for describing job behavior across a variety of tasks and job families). The sets that do not overlap with other job families can be interpreted as relevant to that job family only.

We can translate this knowledge into an index to be used as a heuristic indicator of functional similarity between tasks. Table 2 shows a hypothetical matrix containing functional similarity indices for two TSKs. The cell values (i.e., functional-similarity indices) indicate a perfect relationship between the two action verbs, a small relationship between the object nouns, and a moderate-to-high relationship between the two work aids. Note that the functional similarity for nonequivalent types is undefined. A value could be calculated similar to the functional-similarity value, but the interpretation would be problematic.

Calculating Functional Similarity

The degree of functional similarity between two tasks can be defined as a linear composite of the three individual functional-similarity values calculated over the individual TSK elements. Empirically, we define functional similarity as the number of KSAs that are common to the task statements containing the pair of TSK elements under

Table 2
Functional Similarity Matrix

TSK Elements	TSK Elements		
	Types	Forms	Word Processor
Types	1.0	undefined	undefined
Letters	undefined	.2	undefined
Typewriter	undefined	undefined	.8

consideration. Say we want to determine the functional similarity between the action verbs *completes* and *types*. To calculate an index of functional similarity, we first assemble a list of all task statements containing the action verb *types* and their concomitant KSAs, and a list of task statements that contain the action verb *completes* and their concomitant KSAs. On the basis of these lists, we use the median linkup ratings (expert-generated ratings of the extent to which a knowledge, skill, or ability is required to perform a task) between the KSAs and the tasks, together with the absolute number of KSAs these tasks have in common, to calculate a functional-similarity value for the TSK element pair. The functional-similarity index is the following ratio:

$$h_{fsi} = \frac{\sum_{i=1}^c x_i}{\sum_{i=1}^t x_i},$$

where h_{fsi} is a heuristic indicator of functional similarity, x_i is a median linkup rating between a KSA and its associated task, c is the number of common KSAs, and t is the total number of KSAs for the two TSK elements. This index is bounded by 0 and 1, where 0 indicates that the two elements have no functional similarity and 1 means that there is a complete overlap.

The functional-similarity index is comprised of two pieces of information: (1) the strength of association between the worker attributes and the tasks (via the median linkup ratings); and (2) the absolute number of overlapping KSAs. Note, however, that these indices are comparative heuristics: Meaning is acquired by virtue of our ability to make comparisons between TSK elements.

The following study is intended to show that there are generally applicable rules governing the relationships between tasks and the underlying knowledge, skills, and abilities, and that these relationships can be defined and described in terms of the semantic terminology employed to describe job behavior.

METHOD

Task and Job Sample

Job-analysis data from a state-run mental health facility were obtained for this research. The data had been used to document selection procedures in federal court proceedings. These data consisted of task statements, associated KSAs, and task/KSA median linkup ratings for 27 jobs. Subject-matter experts, supervisors, and job incumbents were used as information sources.

Developing Production Rules

The job-analysis data for the 27 jobs provided a total of 462 task statements. A random process was used to partition these tasks into two groups: (1) tasks employed for the rule database; and (2) a hold-out sample used to evaluate system performance. To ensure that the rules covered a broad area of tasks, mapping rules were based on all but 98 of the tasks. A random process was employed to select the 98 tasks used as the hold-out sample.

A rule database was generated by creating one rule for each of the 364 task statements. For each rule, the antecedent clause contained a task statement represented in TSK form and the consequent clause contained the worker attributes that subject-matter experts had expressed were required to perform the task. The system was implemented in the Lisp dialect, Scheme (Abelson & Sussman, 1985).

The functional-similarity values were calculated using the 364 tasks employed in the rule database. A computer program, implemented in Scheme, was used to calculate the functional-similarity values. Over 28,000 of these values were calculated and employed in the rule database.

System Evaluation

The performance of the system was evaluated by comparing the returned KSAs with the KSAs from the listing generated by subject-matter experts. This created a 2×2 contingency table with four cells: (1) a true-positive cell, for the KSAs returned by the system that were also on the expert-generated listing (i.e., a hit; a correct classification); (2) a false-positive cell, for KSAs that were returned by the system but that were not on the expert listing (a false alarm; an incorrect classification); (3) a true-negative cell, for KSAs that were not returned by the system and that were not on the expert listing (a correct classification); and (4) a false-negative cell, for KSAs that should have been returned but were not (an incorrect classification). This type of design lends itself to evaluation via signal detection theory (SDT; Macmillan & Creelman, 1991). The correspondence design measures the system's ability to discriminate between stimuli (i.e., to detect or discriminate signal from noise). Applied to this case, this means that the design measures the ability of the system to discriminate between those worker requirements that are required of a task and those that are not.

RESULTS

Table 3 presents the result of the mapping run. Of the 98 tasks from the hold-out sample, 59 (60.2%) matched rules constituting partial matches, 36 (36.7%) matched rules constituting exact matches (all three elements matched), and three (3.1%) had no matches (i.e., there were no tasks in the rule database that at least one of the comparable TSK elements matched).

Two statistics used in SDT to summarize performance are the hit rate and the false-alarm rate. The hit rate is calculated as the proportion of true-positives/(true-positives + false-negatives). The false-alarm rate is calculated as the proportion of false-positives/(false-positives + true-negatives). The hit and false-alarm rates were calculated as .74 and .12, respectively.

Two measures commonly used in SDT to measure the sensitivity of a system are d' and c . The *discriminability index*, d' , measures the system's sensitivity to the stimuli (Macmillan & Creelman, 1991), and is calculated as $d' = z(\text{hit rate}) - z(\text{false-alarm rate})$, where z indicates the inverse of the normal distribution function. A d' approximating 1.0 indicates moderate performance; that is, the system is about as likely to generate a hit as it is to generate a false alarm. The value of d' was calculated as 1.82, indicating that the system performed well in terms of its ability to discriminate between those worker attributes that are required for tasks and those that are not.

Table 3
Results of Run

Task Requirement	System Response	
	Required	Not Required
Yes	506	174
No	624	4,514

Table 4
Results With Job-Domain Heuristic

Task Requirement	System Response	
	Required	Not Required
Yes	480	200
No	424	4,714

The measure c (for *criterion*) is a measure of response bias—the tendency of the system to respond “yes” regardless of the actual signal-to-noise ratio and is calculated as $c = -.5 \times [z(\text{hit rate}) + z(\text{false-alarm rate})]$. Positive values of c indicate negative biases, and vice versa. The value of c was calculated as .26, indicating a small negative response bias—that is, a slight tendency to indicate that a particular worker attribute is not required of a task.

Improving the System

A job-domain heuristic was added to the system to reduce the number of false-positives. The addition of job-domain information allows us to acknowledge that jobs from the same job family are more likely to require the same or similar KSAs than are jobs from different job families.

The implementation of a job-domain heuristic required three system modifications. First, based on an examination of the jobs, four job families were created: (1) occupational/psychological adjustment (5 jobs); (2) maintenance (15 jobs); (3) administrative (3 jobs); and (4) pharmaceutical (3 jobs). Second, each knowledge, skill, and ability was classified as either domain specific (42 KSAs) or domain general (18 KSAs). A domain-specific KSA is one that is applicable to a specific job or job family, such as knowledge of pharmacy laws, skill in the use of electrician’s equipment, or knowledge of abnormal psychology. A domain-general KSA is one that is potentially applicable across many jobs and job families, such as communications skill and physical strength. Finally, the system was modified to combine these two information sources into the following heuristic: If the hold-out task and match are from the same job family, return both domain-general and domain-specific KSAs. If the job families differ, return domain-general KSAs only. By returning only domain-general KSAs when job families differ, the number of false-positives should be reduced without adversely affecting the number of true-positives.

Table 4 indicates a decrease of 32% in the number of false alarms from 624 to 424, with only a small reduction—just 5%—for the hits. The difference between the hit and false-alarm cells increased from -108 to $+56$.

The hit rate was calculated as .71 and the false-alarm rate as .08, yielding a d' of 1.93, and c was calculated as .44. This suggests that the addition of the job-domain heuristic increased the sensitivity of the system slightly, whereas the response bias remained about the same.

DISCUSSION

The results support our hypothesis that there are predictable relationships between the terminology employed to describe job behavior and the knowledge, skills, and abilities that are required to perform tasks. The results also suggest that knowledge of these relationships can be codified and implemented in the form of a computerized mapping system.

This research constitutes an incipient attempt to develop a new approach to the task/worker-attribute mapping process. This process attempts to describe explicitly the relationships between tasks and worker requirements. Historically, tasks and their associated requirements have been recognized as related to one another. Little research, if any, has been successful in codifying—in explicitly defining—the relationships between tasks and their underlying knowledge, skills, and abilities (cf. Fleishman & Quaintance, 1984). Of theoretical significance, the present research has provided evidence that there may be generally applicable rules governing the relationships between tasks and the underlying knowledge, skills, and abilities, and that these relationships can be defined and described in terms of the semantic terminology or task lexicons employed to describe job behavior. As with other systems used as intelligent systems in industry (Coovert, 1988), this one should be considered a prototype of a larger, more complex system that can be employed in concert with, or in place of, the traditional mapping process.

REFERENCES

- ABELSON, H., & SUSSMAN, G. J. (1985). *The structure and interpretation of computer programs*. Cambridge, MA: MIT Press.
- BUCHANAN, B. G., & SHORTLIFFE, E. H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project*. Reading, MA: Addison-Wesley.
- COOVERT, M. D. (1988). Analysis and assessment of MIS training needs for naive corporate users. *Association of Computing Machinery SIG-CHI Bulletin*, 18, 79-81.
- CRAIGER, J. P. (1992). *A heuristic procedure for mapping knowledge, skills, and abilities to tasks*. Unpublished doctoral dissertation, University of South Florida, Tampa.
- FINE, S., & WILEY, W. E. (1971). *An introduction to Functional Job Analysis: A scaling of selected tasks from the social welfare field*. Washington, DC: W. E. Upjohn Institute of Employment Research.
- FLEISHMAN, E. A., & MUMFORD, M. D. (1991). Evaluating classifications of job behavior: A construct validation of the Ability Requirement Scales. *Personnel Psychology*, 44, 523-575.
- FLEISHMAN, E. A., & QUAINANCE, M. K. (1984). *Taxonomies of human performance: The description of human tasks*. Orlando, FL: Academic Press.
- GAEIL, S. (1983). *Job analysis: A guide to assessing work activities*. New York: Jossey-Bass.
- KOSKO, B. (1992). *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence*. Englewood Cliffs, NJ: Prentice-Hall.
- LEVINE, E. L. (1983). *Everything you always wanted to know about job analysis*. Tampa, FL: Mariner.
- MACMILLAN, N. A., & CREELMAN, C. D. (1991). *Detection theory: A user's guide*. New York: Cambridge University Press.
- WINSTON, P. H. (1984). *Artificial intelligence*. Reading, MA: Addison-Wesley.
- ZADEH, L. A. (1965). Fuzzy sets. *Information & Control*, 8, 338-353.