A Computational Theory of Complex Problem Solving Using Latent Semantic Analysis

José Quesada, Walter Kintsch ([quesadaj, wkintsch]@psych.colorado.edu)
Institute of Cognitive Science, University of Colorado, Boulder
Boulder, CO 80309-0344 USA

Emilio Gomez (egomez@ugr.es)
Department of Experimental Psychology University of Granada
Campus cartuja, S/N, Granada, Spain

Abstract

Complex Problem Solving (CPS) is a hybrid between field studies and experimental studies. This paper introduces a new, abstract conceptualization of *microworlds* research based on two innovations: (1) a problem representation, which treats protocols as objects in a feature space and, (2) a similarity metric which is defined in this problem space. Latent Semantic Analysis (LSA) is used to analyze performance in CPS, using actions or states as units instead of words and trials instead of text passages. Basic examples of applications are provided, and advantages and limitations are discussed.

Introduction

Many real-word decision making and problem solving situations are (1) *dynamic*, because early actions determine the environment in which subsequent decision must be made, and features of the task environment may change independently of the solver's actions; (2) *time-dependent*, because decisions must be made at the correct moment in relation to environmental demands; and (3) *complex*, in the sense that most variables are not related to each other in one-to-one manner. In these situations, the problem requires not a single decision, but a long series of decisions which are dependent on one another. For a task that is changing continuously, the same action can be successful at moment *t1* and useless at moment *t2*. However, traditional, experimental problem solving research has focused largely on tasks such as anagrams, concept identification, puzzles, etc. that are not representative of the features described above.

In Europe, researchers led by Broadbent (e.g., Broadbent, 1977) in the UK and Dörner (e.g., Dörner, 1975) in Germany, were concerned about that fact and started working on a set of computer-based, experimental tasks that are dynamic, time-dependent, and complex, called *Microworlds*[1]. The study of microworlds is an example of Complex Problem Solving (e.g., French & Funke, 1995).

Compared to traditional Problem Solving, Complex Problem Solving (CPS) radically changed the kind of phenomena reported, the kind of explanations looked for, and even the kind of data that were generated. However, the results obtained to date are far from being integrated and consolidated. This fact led Funke to affirm that 'Despite 10 years of research in the area, there is neither a clearly formulated specific theory nor is there an agreement on how to proceed with respect to the research philosophy. Even worse, no stable phenomena have been observed' (Funke, 1992, p. 25). Almost another 10 years after Funke's argument, although more empirical research has been conducted in the area, we cannot say that the situation has changed drastically. At this moment, there is no theory able to explain even part of the specific effects that have been described or how they can be generalized.

A theory of generalization and similarity is as necessary to psychology as Newton's laws are to physics (Shepard, 1987). However, for CPS there is no common, explicit theory to explain why a complex, dynamic situation is similar to any other situation or how two slices of performance taken from a problem solving task can possibly be compared quantitatively (Klein, Orasanu, Calderwood, & Zsambok, 1993). This lack of formalized, analytical models is slowing down the development of theory in the field. At least two problems make it difficult to apply the classical problem solving approach to CPS, one theoretical and one methodological:

(1) The utility of state space representation for tasks with inner dynamics is reduced because in most CPS environments it is not possible to undo the actions. For example, imagine that two participants in *Firechief* (see below) are in an identical situation (system state) when the trial starts. One of them proceeds to make a control fire on the east side of a fire, while the other one is preparing a control fire on the north front of the fire. After these actions, the system state is no longer identical for them. Now they have to cope with rather different problems. Moreover, if the first participant wants to apply the same technique that

[1] This term sometimes has other meanings. For example, educational applications created to teach physics (Henderson, Klemes, & Eshet, 2000), simulated words in the early AI programs like the block word of SHRDLU, (Winograd, 1972) and static tasks to study decision making (Green, 2001) have been called *Microworlds*. However, we are concerned here only with tasks that fulfill the conditions described above.

the second participant used, there is absolutely no way to come back to the initial state and begin with a new strategy. This situation is not an issue in static tasks like the tower of Hanoi problem because it is always possible to undo a wrong action. Feedback delays (e.g., Brehmer, 1995; Gibson, 2000) and an upsettingly large number of possible states (e.g., Dörner, 1975; Omodei and Wearing, 1995) contribute to the reduced utility of the state space approach.
(2) Traditional methods of knowledge elicitation are not always applicable: Concurrent verbal protocols consistently interfere with performance (Dickson, Omodei & Wearing 2000); measures based on relatedness judgments like rating correlations or pathfinder distance correlations are not sensitive to context manipulations in naturalistic task like fire fighting (Calderwood, 1989).

In this paper we introduce a theory and methodology for CPS tasks based on Latent Semantic Analysis (LSA, Landauer and Dumais, 1997). The theory addresses issues concerning the induction, representation, and application of knowledge. Basically, LSA infers knowledge from the many weak constraints that are present in complex problem solving situations.

LSA does not represent all the possibilities of a system (the system's state space), but only the paths that people have actually followed when interacting with it. This offers a realistic view of how the system is understood and used by humans. LSA is a *computational* theory on how environmental constraints are learned and how they can be described. In terms of Simon's classical parable of *'the ant and the beach'* (Simon, 1981, p. 63), one could say that LSA describes and infers the shape of the beach from the thousands of tracks the ants have left on the beach. In this sense, LSA can be conceived as a computational extension to theories for describing environmental constraints, such as the abstraction hierarchy (Rasmussen, 1985; Vicente, 1999).

LSA has several interesting features that make it a suitable technique for analyzing performance on a complex, dynamic task:
(1) It does not assume independence of decisions; indeed, it uses dependencies between decisions to infer structure. Some methods employed in the past treated CPS performance in a way that assumed that decisions are independent or have short-term dependencies only.
(2) LSA reduces the dimensionality of the space. Imagine a hypothetical problem solving task that, when performed from the beginning to the end, traverses 300 states. Furthermore, let us assume that every state is described by 6 dichotomous variables ($2^6 = 64$ possible states). Since we have 300 states in our sample of performance, there are $64^{300} = 7e541$ possible paths in this task. Every sample would be represented as a matrix of 6 x 300 = 1800 values. With LSA, every sample is represented as a vector of only 100-300 values.
(3) There are no *a-priori* assumptions about '*the beach*'. In most of the analysis performed on *microworld* data the experimenter has to impose some structure (*a-priori*, theoretically driven assumptions) on the data. However, the selection of this theoretical structure (How many strategies are possible? How many strategies are representative? Are they generalizable to different conditions?) can bias the analysis. The LSA approach is self-organizing, and does not require defining an *a priori* theoretical structure, as will be shown below.

Before we start describing what LSA is and how it can be applied to CPS, we would like to stress some abstract considerations that underlie the approach that we are about to implement. These considerations are independent of the procedure itself (other procedures could be defined using this framework), but, in our opinion, an essential step to dealing with the complexity of the tasks at hand: (1) each microworld can be conceptualized as a complex, multidimensional feature space. (2) To address the intractability problem, we usually need to create a representation or transformation of this original multidimensional feature space. To do this, we need to find a set of features that represent the characteristics that make participants different, and to delete those that are not important. (3) Last, each trial of every subject can be conceptualized as an implementation of several values in the feature space. Not only a trial, but every subpart or superpart of a participant's performance (strategies or performance patterns) can be thought of as an object in this space.

We shall illustrate how LSA can be used to analyze CPS tasks, using the *Firechief* microworld as an example.

Description of the example application task

Firechief (Omodei & Wearing, 1995) simulates a forest where a fire is spreading. Participants' task is to extinguish the fire as soon as possible. In order to do so, they can use helicopters and trucks (each one with particular characteristics) that can be controlled by mouse movements and key presses. There are three commands that are used to control the movement and functions of the appliances: (1) Drop water on the current landscape segment; (2) Start a control fire (trucks only) on the current landscape segment; (3) Move an appliance to a specified landscape segment.

Every time a participant performs an action, it is saved in a log file as a row containing action number, command (e.g. drop water or move) or event[2] (e.g., a wind change or a new fire), current performance score, appliance number, appliance type, position, and landscape type. Most of these variables are not continuous, but on a nominal scale, such as type of movement. For more information on the structure of the log files, see Omodei and Wearing (1995).

[2] Events are generated by the system, while actions are generated by the user. Events are also lines in the log file. Only 1-2% of the lines in a log file are events.

The set of trials that was used in this report (referred as *corpus*) was obtained in four experiments described in Quesada, Cañas, & Antoli (2000) and Cañas, Quesada, Antoli & Fajardo (submitted).

Description of LSA

LSA is a machine-learning model that induces representations of the meaning of words by analyzing the relation between words and passages in large bodies of text. LSA is both a method (tool) used to develop technology to improve educational applications, and a theory of knowledge representation used to model well known experimental effects in text comprehension and priming, among others (Landauer & Dumais, 1997). Latent Semantic Analysis was originally developed in the context of information retrieval (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1991) as a way of overcoming problems with polysemy and synonymy. Some words appear in the same contexts (synonyms) and an important part of word usage patterns is blurred by accidental and inessential information. The method used by LSA to capture the essential semantic information is dimension reduction, selecting the most important dimensions from a co-occurrence matrix decomposed using Singular Value Decomposition (see below). As a result, LSA offers a way of assessing semantic similarity between any two samples of text in an automatic, unsupervised way.

LSA has been used in applied settings with a high degree of success in areas like automatic essay grading (Foltz, Laham, & Landauer, 1999) and automatic tutoring to improve summarization skills in children (E. Kintsch, Steinhart, Stahl, Matthews, Lamb, & the LSA Research Group, 2000). As a model, LSA's most impressive achievements have been in human language acquisition simulations (Landauer & Dumais, 1997) and in modeling of high-level comprehension phenomena like metaphor understanding, causal inferences and judgments of similarity (Kintsch, 2001).

Although LSA has been mostly used on text corpora, our basic point is that LSA can be applied to any domain of knowledge where there are a high number of weak relations between tokens, as in CPS log files. Instead of word usage statistics obtained from huge samples of text, we have used a representative amount of activity in controlling dynamic systems, and actions or states have been used to develop an objective measure of similarity in the changing, time-dependent, highly complex experimental tasks known as *microworlds*. The next sections show the basic steps involved in this analysis and presents some examples of the powerful results that can be obtained thereby.

LSA applied to Microworlds

LSA starts with the creation of a matrix of actions by trials. Note that this is not an exhaustive state space, or a mapping of all possible transitions between actions, since in most of the systems – other than small ones like Hayes & Broadbent's (1988) sugar factory and the like - this task would be excessively demanding (see Buchner, Funke & Berry, 1995). Our corpus was composed of 360,199 actions in 3441 trials. Among them, only 75,565 were different actions, which means that on average each action appears 6.25 times in the corpus. Note that we are representing *only* the information that actual people interacting with the system experienced, not all possible actions in this microworld.

Each of these 75,565 rows stands for a unique action, and each of the 3441 columns stand for a trial. Each cell contains the frequency with which the action of its row appears in the trial denoted by its column. Note that most of the cells will contain a frequency of zero, since most actions appear in only a few trials and not in the rest.

This matrix of frequencies is decomposed using Singular Value Decomposition (SVD). Any matrix can be decomposed and then recomposed perfectly using only as many factors as the smallest dimension of the original matrix. However, an interesting phenomenon occurs when the original matrix is recomposed using fewer dimensions than necessary: the reconstructed matrix is a least-squares best fit. When the actions-by-trials matrix is recomposed using a small fraction of the available dimensions (usually between 100 and 300 dimensions), the new matrix contains information that has been inferred from the dependencies between actions and the context where these actions appeared. In fact, the contexts where these actions did not appear are as important - carry as much information - as those where they did. The microworld is a new multidimensional feature space, where both actions and context (trials) are represented in a way that amplifies those characteristics that make participants different, and delete those that are not important for classifying their performance.

Some examples of possible analysis

LSA allows us to measure the functional similarity between actions in CPS tasks. Some actions can be considered as *functional synonyms*: they appear in the same contexts, and fulfill approximately the same function. The following example illustrates this idea.

Table 1: Example of how LSA captures similarity at a molecular (action) level

	Time *t1*	**Time *t2***
Example 1	move_11_9_forest	Drop_11_9_forest
Example 2	move_15_15_forest	Drop_15_15_forest
Example 3	move_10_9_forest	Drop_10_9_forest
Example 4	move_11_9_forest	control_11_9_forest

In Table 1, four different actions some actions are shown. For simplicity, some variables that are normally contained in the log files have been removed. Example1 contains a movement to the point (11, 9) in the screen, which is of type *forest,* and then, a drop water action there. Example 3 shows a very similar picture, where the movement is done to a contiguous cell (10,9) that is also of type *forest*. From a human point of view, these two examples are highly similar. For LSA they are too, as can be seen in their similarity expressed as a cosine of 0.854 in Table 2.

Table 2: Similarities between Table 1 examples (cosines).

	Example 1	Example 2	Example 3	Example 4
Example 1		0.124	0.854	0.662
Example 2			0.1259	0.077
Example 3				0.566
Example 4				

The second example has a rather different meaning since the cell targeted is (15,15), quite far from the cell used in examples 1 and 3. The cosines between them and example2 (.124 and .125) are, accordingly, smaller than the one between 1 and 3.

Example 4 describes an action that has been performed in the same cell as in example 1 (11,9), but this time is a control fire instead of a drop-water action. The cosine between 1 and 4 is high (0.56), expressing a certain similarity between the two actions, but not as high as in examples 1 and 3, where the objective similarity is more evident.

Tables 3 and 4 present a more complex example where wider slices of performance (8 actions) are compared. The samples labeled Example1, Example2, and Example 3 are beginnings of trials that have been selected randomly from the *corpus*. This time, all the usable information contained in the log file is displayed. Each action has six components: type of action, appliance number, appliance type, departure cell, arrival cell and type of arrival cell.

One difficulty arises. When LSA is used on text, cosines are easily understood since every reader has an intuitive experience of meaning (e.g., the sentences 'The man was driving a yellow car' and 'The man was traveling in a red car' have a cosine of .89, and our common sense tells us that these sentences convey similar information). When LSA is used on samples of performance from a *microworld*, there is no way the reader can understand the meaning of the log files without watching a replay or having an extraordinarily vivid imagination plus experience with the task. For most people, the following extracts in Table 3 are hardly understandable. For researchers familiar with *Firechief*, they should be as clear as a piece of sheet music to a musician. However, understanding the contents of these examples is not *conditio sine qua non* for understanding the advantage of LSA analysis over two other methods, namely exact matching and correlation between transition matrices. Suffice it to say that Examples 1 and 2 are very similar and Example 3 is very different from them. The attentive observer could induce this from the locations (coordinates in the Firechief map), the type of actions, and type of landscape cell. An *exact matching* method would count the number of times that the same action occurs in two examples. Then, the number of matches divided by the total number of actions in the example provides a measure of the similarity between two samples. This method would render a similarity of 1/8 between example 1 and 2, and zero in comparisons 1 vs. 3 and 2 vs. 3. This method is equivalent to keyword counting in text, which is known to be incapable of capturing similarities in meaning, because of the polysemy and synonymy effects discussed above.

A somewhat more flexible method is the use of *transitions between actions*, as proposed by Howie and Vicente (1998) and used in Quesada et al. (2000) and Cañas et al. (submitted). It is based on counting the number of times that one type of action precedes any other type. The frequencies of every transition are registered in cells in a table, and then the resulting tables for two examples are correlated. The method cannot account for all the variability in actions, because of the huge amount of zero entries that artificially

Table 3**:** First 8 movements in 3 slices randomly sampled from the *Firechief* experiments described in Quesada et al. (2000) and Cañas et al. (submitted). When an action is shared by two extracts, it is marked as a shaded cell.

Example 1	**Example 2**	**Example 3**
move_2_truck_4_11_13_3_forest	move_2_truck_4_11_12_15_forest	move_2_truck_4_11_2_2_pasture
move_1_truck_4_14_16_14_forest	move_1_truck_4_14_13_5_forest	move_1_truck_4_14_0_5_forest
move_3_copter_8_6_11_12_forest	move_4_copter_11_4_11_9_forest	move_4_copter_8_6_8_4_clearing
move_4_copter_11_4_11_9_forest	drop_water_4_copter_11_9_forest	move_3_copter_8_6_8_10_clearing
Control_fire_2_truck_13_3_forest	move_4_copter_11_9_13_8_forest	control_fire_2_truck_2_2_pasture
Control_fire_1_truck_16_14_forest	control_fire_2_truck_12_15_forest	control_fire_1_truck_0_5_forest
move_2_truck_13_3_17_7_clearing	move_2_truck_12_15_13_14_forest	move_4_copter_8_4_4_2_forest
move_1_truck_16_14_20_12_forest	control_fire_2_truck_13_14_forest	move_3_copter_8_10_2_3_clearing

increase the correlation, so only action type was considered. This analysis is shown in tables 4(*a*,*b*,*c*). Since lots of information contained in the log files has been dropped, the method does not distinguish between these examples. The correlation between table *a* and *b* is 0.971; exactly the same correlation is obtained for tables b and *c*, and the comparison between *a* and *c* is 1 since the sequence of *type of action* is exactly the same. Thus, this method is seriously flawed because it yields implausible similarity estimates.

Finally, let us look at the results of similarity estimation using LSA cosines. The vector representing the sample has been calculated as the average of the 8 action vectors. Example 1 vs. example 2 has a cosine of 0.721, a high similarity value. Even though these samples share only 1/8 of the actions, LSA has correctly inferred that the remaining actions, although different, are functionally related. Comparisons between 1-3 and 2-3 have cosines as low as 0.050 and 0.071 respectively, showing that these action sequences are different indeed.

Correlations between LSA and Human Judgment

More formal comparisons between the performance of LSA and human observers than mere plausibility judgments are also possible. The problem is that, contrary to what happens when one uses LSA to model text comprehension, it is not easy to find experts in the task at hand. Everybody is a good example of the expert reader, but few people are expert in controlling the particular dynamic system called *Firecshief*. To test our assertions about LSA, we recruited 3 persons and gave them extended practice, so they could learn the constraints of the task.

After 24 practice trials, these participants were used to assess the external validity of LSA similarities. Using *Firechief's* replay option, participants had to watch 7 pairs of trials (at a pace faster than normal) and express similarity judgments about these pairs. People watched a randomly ordered series of trials, in a different order for each participant, which were selected as a function of the LSA cosines (pairs A, B, C, D, E, F, G with cosines 0.75, 0.90, 0.53, 0.60, 0.12 and 0.06 respectively). One of the pairs was presented twice to measure test-retest reliability. That is, for example, pair G was exactly the same as pair A for one participant, the same as pair F for another participant, etc. All the possible stimulus pairs were presented to each participant. Participants had to answer which pair seemed more similar. For example, LSA would say that pair B is more related than pair C, since the cosines are 0.90 and .53 respectively.

LSA cosines predicted human similarity judgments quite well. For 3 participants in this pilot study, the proportion of agreement LSA-human was 6/19, 14/19, and 13/19 respectively. Participants with strong agreement with LSA also showed more consistency in their judgments, that is they answered to the repeated item in the same way. The participant who had low agreement with LSA had performed poorly on the repeated item, which suggests that she may not have learned enough about the task or was not paying sufficient attention. Even so, the average agreement between LSA cosines and human judgments was 0.57, far superior to the agreement expected by chance, 0.5 ^ 19 = 2e-5.

Conclusions

LSA seems to be a promising new way of approaching Complex Problem Solving performance that overcomes some of the known limitations of previous methods. Apart from the features listed in the introduction, there are some additional pragmatic LSA advantages worth noting: (1) Since the basic unit of analysis is the token (action or state), even systems that are described in terms of nominal (discrete) variables can be analyzed. Both actions and states can be used as units. (2) The semantic matching mechanism permits discovery of similarities beyond simple coincidence in the log files containing actions or states. That is, participants who are using different interventions to realize the same strategy will be considered similar even if their log files share no actions (or states). (3) The level of granularity (whether we are working with individual tokens, slices of performance, whole trials, or collections of trials) need not be defined *a-priori*. Since every object, from one token to the participant's whole performance, can be represented as a vector in the high-dimensional problem space constructed by LSA, analyses can be performed at any level of detail.

There are, however, a number of limitations to the proposed method: (1) A huge sample of data is needed to construct

Table 4: Transitions between actions considering type of action only as described in Quesada et al. (Quesada et al., 2000) and Cañas et al. (submitted), for the examples 1,2 and 3. Cells contain frequencies of the transition defined by its row and its column. For instance, the number 4 in the center cell in table 4a means that in example 1 the transition move-move has appeared four times.

(a)

	Example 1		
	drop	**move**	**Control**
Drop	0	0	0
Move	0	4	1
Control	0	1	1

(b)

	Example 2		
	drop	**move**	**control**
Drop	0	1	0
Move	1	2	2
Control	0	1	0

(c)

	Example 3		
	drop	**move**	**control**
drop	0	0	0
move	0	4	1
control	0	1	1

the problem space. (2) Order effects are not taken into account. This means that, for LSA, a trial where the tokens have been scrambled to a random order has exactly the same meaning as the original version This is a serious but, as we have shown, not a fatal limitation, as long as LSA is used with care in CPS tasks. (3) Though the SVD analysis is common practice and can be found in several statistical packages, a powerful computer is needed to run large analyses.

Acknowledgments

Our acknowledgements to Tom Landauer for proposing interesting issues concerning the selection of the unit of analysis in Complex Problem Solving. We are grateful to Kim Vicente and John Hajdukiewicz for sharing experimental data and insightful discussions. Many thanks to Bill Oliver, who provided passionate methodological discussions and theoretical contributions.

This research was in part supported by Grant EIA – 0121201 from the National Science Foundation.

References

Brehmer, B. (1995). Feedback delays in complex dynamic decision tasks. *In P. Frensch and J. Funke, (Eds.) Complex Problem Solving: The European Perspective. Hillsdale, NJ: Lawrence Erlbaum*.

Broadbent, D. E. (1977). Levels, hierarchies and the locus of control. *Quarterly Journal of Experimental Psychology, 32*, 109-118.

Buchner, A., Funke, J., & Berry, D. (1995). Negative correlations between control performance and verbalizable knowledge: Indicators for implicit learning in process control tasks? *Quarterly Journal of Experimental Psychology, 48A*, 166-187.

Calderwood, R. (1989). The role of context in modeling domain knowledge. *Unpublished doctoral dissertation, University of New Mexico*.

Cañas, J. J., Quesada, J. F., Antolí, A., & Fajardo, I. (submitted). Cognitive flexibility and adaptability to environmental changes in dynamic complex problem solving tasks.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R. (1991). Indexing By Latent Semantic Analysis. *Journal of the American Society For Information Science, 41*, 391-407.

Dickson, J., McLennan, J, Omodei, M. M. (2000). Effects of concurrent verbalization on a time pressured dynamic decision task. *Journal of General Psychology, 127*, 217-228.

Dörner, D. (1975). Wie Menschen eine Welt verbessern wollten und sie dabei zerstörten (How people wanted to improve the world but destroyed it). *Bild der Wissenschaft, Heft 2*.

Foltz, P. W., Laham, D., & Landauer, T. K. (1999). The Intelligent Essay Assessor: Applications to Educational Technology. *Interactive Multimedia Education Journal of Computer enhanced learning On-line journal, 1(2)*.

Frensch, P., & Funke, J. (1995). Complex Problem Solving: The European Perspective. Hillsdale, NJ: Lawrence Erlbaum.

Funke, J. (1992). Dealing with dynamic systems: Research strategy, diagnostic approach and experimental results. *German Journal of Psychology, 16*, 24-43.

Gibson, F. P. (2000). Feedback delays: How can decision makers learn not to buy a new car every time the garage is empty? *Organizational Behavior and Human Decision Processes, 83*(1), 141 - 166.

Green, D. W. (2001). Understanding microworlds. *Quarterly Journal of Experimental Psychology, Section A-Human Experimental Psychology, 54*(3), 879-901.

Hayes, N. A., & Broadbent, D. E. (1988). Two modes of learning for interactive tasks. *Cognition, 28*(3), 249-276.

Henderson, L., Klemes, J., & Eshet, Y. (2000). Just playing a game? Educational simulation software and cognitive outcomes. *Journal of Educational Computing Research, 22*(1), 105-129.

Howie, D. E., & Vicente, K. J. (1998). Measures of operator performance in complex, dynamic microworlds: Advancing the state of the art. *Ergonomics, 41*, 85-150.

Kintsch, E., Steinhart, D., Stahl, G., Matthews, C., Lamb, R., & the LSA research Group (2000). Developing summarization skills through the use of LSA-backed feedback. *Interactive Learning Environments, 8*(2), 87-109.

Kintsch, W. (2001). Predication. *Cognitive Science, 25*, 173-202.

Klein, G. A., Orasanu, J., Calderwood, R., & Zsambok, C. E. (Eds.). (1993). *Decision making in action: Models and methods*. Norwood, NJ: Ablex Publishing Corporation.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 211-240.

Omodei, M. M., & Wearing, A. J. (1995). The Fire Chief microworld generating program: An illustration of computer-simulated microworlds as an experimental paradigm for studying complex decision-making behavior. *Behavior Research Methods, Instruments & Computers, 27*, 303-316.

Quesada, J. F., Cañas, J. J., & Antoli, A. (2000). An explanation of human errors based on environmental changes and problem solving strategies. In P. Wright & S. Dekker & W. C.P. (Eds.), *ECCE-10: Confronting Reality*. Sweden: EACE.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(2), 234-243.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science, 237*, 1317-1323.

Simon, H. A. (1981). *The sciences of the artificial*: MIT press.

Vicente, K. J. (1999). *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-based work*. London: Lawrence Erlbaum Associates.

Winograd, T. (1972). A procedural model of language understanding. In R. Schank, Colby, K.(Ed.), *Computer models of thought and Language*. Sand Francisco: W.H. Freeman.