Editor, Human Factors
Human Factors and Ergonomics Society
1124 Montana Ave., Suite B
Santa Monica, CA 90403

Dear Editor of HUMAN FACTORS,

Please consider "Automatic, human-like representation of task constraints" for publication in Human Factors, in the category "review articles". The manuscript contains 11575 words of main text in 35 pages (double-spaced, APA 5th ed. format).

I think the paper will be of interest to the human factors/ergonomics community, and may foster discussion if published in the journal. I think it presents new ideas that may be considered both theoretical and methodological contributions to the field.

Thanks a lot in advance for your time.

Sincerely,
Jose Quesada, PhD.

jquesada@andrew.cmu.edu
http://lsa.colorado.edu/~quesadaj
http://www.andrew.cmu.edu/~jquesada
Porter Hall
office PH208-J
5000 Forbes Ave.
15213, Pittsburgh, PA

Research associate
Dept. of Social and Decision Sciences
Carnegie Mellon University
Phone: 412 268 6011
Fax: 412 268 6938

Running head: cognitive representation of tasks

Automatic, human-like representation of task constraints

Jose Quesada[1], Walter Kintsch[1], Emilio Gomez[2]

[1]Institute of Cognitive Science, University of Colorado

[2]Department of Experimental Psychology, University of Granada, Spain

Submitted to: Human Factors

Abstract

Current approaches to complex, dynamic tasks require the experimenter to decompose and analyze a priori the components of the task (cognitive task analysis). Although the literature is prolific on how to parse tasks in cognitively plausible ways, methods to date are not well specified, and different researchers using the same method on the same task can come up with different descriptions. As a solution to these problems we propose Latent problem Solving Analysis (LPSA), a computational theory that generates human-like representations of the tasks by analyzing large numbers of logged trials of virtually any task. This approach does not rely on verbal protocols, similarity judgments, expert knowledge elicitation techniques, or analytical descriptions of the task such as manuals to create the representation.

Keywords: Problem Solving, Knowledge representation, Complex systems, Computer simulation

Automatic, human-like representation of task constraints

Introduction

Cognitive science is the study of the interaction between the cognitive system and the environment, or task. We know quite a lot about how the cognitive system works. For example, we know that there is a limited working memory capacity, that our attentional system cannot track more than a certain number of mobile targets with 100% accuracy, that some cognitive activities can be entertained simultaneously without affecting each other whereas others cannot, etc. But what do we know about the task? In the literature, this area seems to be covered by Cognitive Task Analysis (CTA). However, the state of our knowledge of the task pales in comparison to our knowledge of the cognitive system. This is a very peculiar situation, since many authors (e.g., Brunswik, 1956; Juslin, Olsson, & Winman, 1996; Simon, 1981; Vicente, 1999) have proposed that the environment is at least as important as the individual's cognitive system in explaining behavior. Why do we not have comparable theories about the environment?

It is certainly not because of a lack of effort or interest from the scientific community. CTA is the area of cognitive science that is concerned with how people represent their natural work environments, particularly the really complex ones. The applications of CTA are very important and necessary in contemporary highly technical jobs; among them are training, user interface improvement, and prediction of dangerous human-computer situations including human errors. Hoffman and Woods (2000) pointed out that there is currently an explosion of acronyms and techniques for the study of cognitive systems in context: "different purposes, histories, and threads have resulted in a fragmented landscape, parochialism, imperialism, and proliferation of labels" (p. 1). Schraagen, Chipman and Shute (2000) review 20 papers, and each of those presents on average several dozen techniques for task analysis. These techniques include verbal

protocol analysis, interviews, elaborated knowledge elicitation methods, similarity judgments, analytical descriptions of the task such as manuals, etc. However, there are no integrated theories about the task, nor are there consistent criteria for accepting or rejecting a particular task analysis. Among these methods there are two common factors that seem particularly extended and accepted by the community. In what follows we criticize these two and propose an alternative solution.

The first methodological issue we want to concentrate on is experimenter introspection. Kieras and Meyer (2000) argued that most of the analyses performed in the literature respond to a certain amount of experimenter introspection: “the methods are nothing more, nor less, than guides to make the analysis process more systematic, dependable, and communicable, but the analyst’s intuition and expertise is still the actual source of the user task” (p. 256-57).

Certain modelers tend to analyze tasks in a way that is mostly dictated by their intuition and understanding (educated guesses) of how people proceed when interacting with the environment. For example, Lee and Anderson (2001) used a simplified Air Traffic Controller task (ATC) in a controlled lab situation; they used three levels of description: (a) the unit-task level, (b) the functional level, and (c) the keystroke level. However, Lee and Anderson (2001) used no learning or representation theory to generate such decompositions: "[Our] task decomposition does not depend on any particular theory of skill acquisition. Rather, it solely relies on a pretheoretical task analysis" (p. 273).

It seems that attributing one’s decomposition of a task to a “pretheoretical task analysis” justifies anything one wants to say about the task. This is in part because experimenters are often experts in the task they use, and nobody is interested in questioning their expertise.

Another example is the work on the thermodynamic microworld Duress by Vicente and colleagues (e.g., Christoffersen, Hunter, & Vicente, 1996, 1997, 1998). Vicente and collaborators have developed a process measure (Yu, Lau, Vicente, & Carter, 1998) of control performance in complex, dynamic situations based on the Abstraction Hierarchy (AH). The AH is an analytical tool that has been proposed to decompose the variability of complex work domains (Hajdukiewicz & Vicente, 1999; Rasmussen, 1985). Each level of the AH is a model of the same work domain, but the levels differ in abstraction and resolution (part–whole granularity). Thus, each level relies on a different set of attributes or "language" (Yu et al., 1998). The defining feature of this hierarchy is that its elements in different levels are connected by means-end links. The AH for Duress contains the following levels: (1) Functional purpose: objects at this level of abstraction correspond to overall system goals. (2) Abstract function: this level can be described in terms of the conservation of mass and energy for each reservoir subsystem. (3) Generalized function: this level describes flows and storage of heat and water. (4) Physical function: this level describes the states of system components. The Abstraction Hierarchy proposed for Duress is not directly related to any cognitive theories, and that certainly harms the generalizability of the results obtained with it to other tasks. If we had theoretically well-motivated analyses of the tasks, we could say that tasks A and B both depend on process P, so the results from both tasks are comparable.

If the structure for assessing similarity is built on the intuitions of an experimenter, then the task analysis will not be very reliable or generalizable, in the sense that two or more researchers could come up with different decompositions. A possible way out of this impasse is to use the verbalizations of the actual experts in the task instead of the experimenter’s intuition. This proposal has been called “analysis of verbal protocols” and has a very long tradition in

psychology and human factors. But this method, too, has flaws. Since humans are not completely transparent to introspection, there is virtually no chance that a concurrent verbalization can be used as an exhaustive description of the processes that take place during any particular event in our conscious life. Many authors have doubted the validity of verbal protocols as data to test theories of mental behavior (e.g., Bainbridge, 1999), which is exactly the purpose of CTA: to understand the cognitive activities that take place during the task. Moreover, it has been shown that people interacting with complex, dynamic tasks like most of the ones that are of interest for the human factor practitioner are severely affected by concurrent verbalization (Dickson, McLennan, & Omodei, 2000).

In this paper, we present the researcher's intuition as the main bottleneck in task analysis. This idea is in agreement with Kieras and Meyer (2000), who advocated a more formal approach to task analysis based on computationally implemented cognitive architectures. Starting with a task analysis is common not only in applied contexts, but in laboratory tasks too. We focus the analysis presented in this paper on microworlds, but it would be easy to generalize to other tasks, both purely laboratory ones and applied contexts.

This article has four parts. First, we introduce the problem: theories about the task are based on decompositions done primarily using intuition and introspection. Second, we show that a bottom-up approach using corpora of logged activity is a valid alternative to using intuition, introspection, and verbal protocols. We propose a possible implementation of this solution in Latent Problem Solving Analysis (LPSA), a computational theory of representation in complex, dynamic problem solving (Quesada, Kintsch, & Gomez, 2003a, 2003b, submitted-b). Third, we present an example of use of LPSA on a well-known microworld, Firechief (Omodei & Wearing, 1993, 1995). Last, we present our general discussion and conclusions.

Common assumptions adopted to generate theories about the task

In the three following sections we criticize three assumptions used to generate theories about the task: (1) task decomposability, (2) intuition as the main tool, and (3) strategies and heuristics.

Task decomposability

CTA implicitly assumes that the best way to understand a human performing a complex task is to decompose that task into smaller components that will then be analyzed again, until some kind of basic "cognitive alphabet" or set of "cognitive primitives" is discovered. But it is questionable that we can identify components small enough to be common to all tasks. This path leads to a pretty radical (and questionable) assumption: the cognitive system can be described with a limited vocabulary of task-independent P processes. Thus there exists the possibility of a research program of "mapping" the cognitive system as we map the human genome. Several theoreticians have reasons to believe that this path is not justifiable, and some others have shown commitment to the idea but never started the project of generating the common vocabulary to describe any task in cognitive terms. Current CTA techniques do not aspire to generate this common alphabet; they seem to be content with generating subunits that are valid for only one task. This situation echoes similar decisions made by researchers in the decision-making and problem-solving communities.

Newell and Simon (1972) proposed that humans, when engaged in tasks such as solving cryptarithmetic problems, proving theorems, and playing chess, are representable as information processing systems. They also proposed that one of the characteristics of information processing systems is that they work on a (fixed) set of elementary information processes (EIPs). However,

in their work they did not explicitly address the issue of generating a complete set of EIPs that could be used to describe any task.

Payne, Bettman, and Johnson (1993) used the same idea of elementary information processes to describe their decision-making situations. Their paradigm consists of a screen where several decision alternatives, described using few differentiating features, are presented covered. The actions of the participant (mouse movements) will show the value of a particular feature of a particular alternative. Under limited time, people have to sample the information available and come to a decision. In this situation, Payne et al. proposed a limited set of EIPs, but these were not intended to be generalizable to other situations.

The goal of cognitive architectures such as ACT (e.g., Anderson & Lebiere, 1998) and SOAR (e.g., Newell, 1990) seems less to create a basic, exhaustive vocabulary either, but more to generate a catalog of cognitive constraints that have been derived experimentally (e.g., working memory limitations, attention span, etc.).

Gigerenzer and colleagues (e.g., Gigerenzer & Goldstein, 1996) constructed a theory where the basic concepts and mechanisms are attuned to a particular environment, instead of being general to any situation. They opposed the idea that analysis of the task could generate context-independent components.

The implicit assumption of "complex task decomposability" has deep consequences. Researchers have had to face complex phenomena in many other areas of cognitive science. In this situation, the two obvious ways to proceed are decomposition-based approaches and holistic models. In most cognitive science fields, researchers have explored both paths. However, decompositions are not as successful as holistic approaches in some cases—for example, object recognition and the cognitive representation of word meaning.

In object recognition, Biederman (1987) asserted that there should be some set of basic units that could be used to decompose any natural concrete object. He developed a set of 36 basic units called geons. The basic form was a generalized cone. Object recognition could then be reduced to a parsing problem where the cognitive system decomposes any complex form to those basic components. Unfortunately, this structural decomposition approach proved to be affected by serious issues: for example, it is mathematically equivalent to labeled graph matching, which is a difficult combinatorial problem that is computationally intractable (Garey & Johnson, 1979).

An alternative is to consider that the cognitive system never decomposes objects into parts. This has the advantage that we do not have to define a basic vocabulary beforehand. Edelman (1998; 1999) took this approach in his ”chorus of prototypes” theory. He proposed that the cognitive system creates a shape space where objects are represented. The space is low-dimensional, consisting in specialized modules each attuned to a certain form (e.g., ”car,” ”dinosaur,” ”human,” plane,” etc.) that will fire proportionally in their presence. A new object will be similar to those prototypes, and then evoke a particular response from them (the chorus sings). The theory bypasses the problems described for structural decomposition theories (e.g., necessity for a vocabulary, combinatorial explosion), and offers interesting solutions to the problem of representation.

In the study of the cognitive representation of word meaning, some theories (e.g., Schank, 1972) proposed that meaning could be analyzed and decomposed into units that would work as chemical elements. These unitary concepts could be combined using a set of rules that specify which types of concepts can depend on which other types, as well as the different kinds of dependency relationships between concepts.

The research program described by Schank (1972) would aim to generate the equivalent of the periodic table for the cognitive representation of word meaning. These theories never were materialized and were replaced by theories that define the meaning of a word as a function of the contexts in which it appears (e.g., LSA, Landauer & Dumais, 1997, see the dedicated section below).

When it comes to understanding tasks, however, for some reason practitioners and researchers alike have assumed that the decomposition of complex activities into smaller units is the best way to go when they need to work with complex tasks, leaving the alternative path unexplored. In this paper we argue that a corpus-based approach, where decomposition is not needed to represent work domains, can be used effectively in the study of problem solving.

Intuition as the main tool

The problem with task decomposability is increased by the belief that it has to be guided by the experimenter's common sense. In some cases, the task analysis is assisted by a formalism, but even then the implicit assumption is that the experimenter's intuition is the best way to understand the task.

This is an additional assumption that is independent of the previous one: experimenters could agree that task decomposition is necessary for scientific progress, and use methods other than common sense or intuition to partition the task. However, these two assumptions are often linked: most people who advocate task decomposition also use their understanding of the task to do the partitioning.

Strategies and heuristics

On many occasions, the experimenter needs to propose a strategy or heuristic that people follow when interacting with the task. This problem is related to the first two in that this is usually done

by using just the common sense of the experimenter, or knowledge elicitation techniques that tap mainly introspection processes from experts. Our criticism is directed not to the use of intuition or common sense to generate theories, but to the abuse of it when other more plausible, objective methods of generating explanatory factors are available. It is clear to us that a top-down component is needed to advance science, and that introspection and intuition have a capital importance in this process; however, we believe that they should be used when no alternative inference methods can warrant a more systematic abstraction from the data.

Most of the task descriptions in the literature assume that participants use strategies in their actions, and this is a powerful explanatory factor in the theory. But how do we know what are the strategies that could possibly be used? How many can we generate? How can we be sure that the ones we propose are an exhaustive set, or at least representative of the population of possible ones? The only answer is the belief that the experimenter knows the task well and his or her intuition can be used as a gold standard.

In other cases, the researcher uses the assumptions of a cognitive architecture to help identify the task strategy. Kieras and Meyer (2000) presented this idea in a very clear statement:

> Typically, the researcher makes an intuitive guess about what the task strategy is and then sees whether the predictions generated from the assumed strategy and architecture fit the data. If not, the researcher modifies the strategy or architecture and repeats until a good fit is obtained between the data and predictions (p. 241).

The problem in this case is that one can never be sure where the source of disagreement is—in the architecture or in the strategy. This is so because on most occasions there is more than one possible strategy that fits the dataset in a satisfactory way, as well as more than one plausible modification of the architecture that could accommodate the dataset well enough.

A second concern of ours is that strategies are most of the time task-specific. These strategies are difficult to generalize outside the domain where they were created because they depend heavily on the semantics of their domain, making it very difficult to construct a theory on strategies.

Experimenters could propose generalized strategies that are independent of the context of application. An effort to catalog such strategies has been made by Gordon (2003). However, this creates an epistemological problem, since ”these strategies are no longer context-specific, and thus, cannot be verified directly by empirical data” (Xiao & Vicente, 2000, p. 98).

A possible solution: A bottom-up approach to deal with complexity

Is there any way to escape the issues raised? We propose that a possible solution is to use a mainly bottom-up approach to task representation. The basic idea is to collect a representative sample of human activity in the task we are interested in, and use this information to infer the task constraints and represent them. This sample is called a corpus. We assume that this task has to be performed by the cognitive system in the wild, using the information available in the environment with a minimum of pre-built knowledge.

The data of choice for our approach is the log file of completed activity: a trial in a laboratory task, a complete cycle (for example, an airplane landing) in a naturalistic task. We define context using this log file as a unit, and our similarity metric is built on top of this decision. The segmentation of problem solving-activity into trials seems as natural as dividing a corpus of text into documents.

We propose that a theory of the task should be based on the constraints that can be inferred from the interaction between the human cognitive system and the environment. Instead of using the experimenter’s common sense, we propose to use rational principles to generate the

representation. The question we ask is: Given this environment, what kind of representation can a cognitive system build that maximizes its ability to correctly generalize and predict future states?

The abstraction performed should be optimal in the particular environment. It is very difficult to propose a mechanism for inference and generalization using a corpus that is optimal for every environment, because environments are very varied and it is difficult to imagine a formalism that is flexible enough to capture the structure of very different environments.

Another important concern is the validity of the formalism. The structure generated by the mechanism should reflect how people represent their environment in a valid way. However, testing this validity is difficult because we cannot use direct measures of the structure, but only indirect ones. We propose two ways in which we can test the validity of the psychological space generated: (1) human similarity judgments, and (2) prediction. Human similarity judgments are widely used in cognitive science as a source of data to validate theories of representation. They are reliable in the sense that people tend to emit very consistent similarity judgments when the context is kept constant. They are also easy to get and intuitively appealing. Prediction plays a very important role in humans' interaction with the environment. Some scientists argue that many features of the cognitive system (such as representation, memory, and categorization) can be conceptualized as tools that help to predict the next states of an organism's environment (e.g., Anderson, 1990).

We have used similarity judgments and prediction tasks to validate the representations generated in our model, and they are used as evidence in the section about LPSA below.

Our emphasis in presenting this framework presented is ”to find out how far a mostly bottom-up approach to representation can be taken” (Edelman, 1998 p. 459). Notoriously, this is usually far beyond what most people would have expected.

In the rest of the paper, we focus on microworld research because it is easy to design experimental situations of interest, we can use undergraduates as participants, and there are ample datasets already available. However, we think the proposal that we present is valid for any kind of complex activity that can produce log files.

Context-based similarity

This section is concerned with how to proceed once we accept that the log-file information is what we are going to use in our analysis. We present Latent Problem Solving Analysis, our main solution for representation of complex tasks in a way that resembles human representation. LPSA proposes a working definition of similarity based on the contextual usage of types (that is, similarity is context based). Types are defined as all unique actions or states in a sample of human-system interaction known as the corpus. Tokens are the particular occurrences of each of these types. We will not take a position on the distinction between logged actions and states, since some task domains require one and some another. The approach that we present can work with either. We simply assume that a log file consists of a list of actions or states that we call tokens. The total number of types in a sample of behavior that is considered representative of a task domain is termed its vocabulary size. Complex, dynamic tasks such as the ones that interest the CTA community usually have large vocabulary sizes, comparable to the vocabularies of natural languages such as English.

The most obvious way to compare two slices of performance (e.g., trials) is to count their overlap in terms of tokens. However, this approach will fail soon because in complex, dynamic

tasks the average frequency of a type is very low, and most performance trials will share no tokens. We can imagine a large matrix where the rows are all types in a corpus of activity sampled from a task domain, and the columns are different trials at the task, by many different people. Most of the cells in this matrix will be zero. This is a characteristic of multidimensional spaces that has been termed the curse of dimensionality: high-dimensional spaces are inherently sparse. Comparing the vectors for these trials with such a sparse dataset will show high correlations just because of the shared zero cells, but these correlations will be artifactual and will not capture the similarity relations between the trials compared. In this situation, the experimenter may try to reduce the vocabulary size (and the sparsity of the matrix) by selecting only a small subset of the sources of variance in a type. For example s/he may decide that only the color of a moving target on a screen, and not other features like position, shape, direction, speed, etc., is relevant for the task that the operator performs. These reductions imply forcing the dataset into a hypothesized structure, and are normally performed a priori using the intuition of the experimenter. In the next section we describe an example of this situation, using a published study that faces these problems (Cañas, Quesada, Antolí, & Fajardo, 2003), and propose our solution. Our view is that there is a more effective way of dealing with this situation without having to impose any structure on the data, and we have instantiated it in LPSA, which we explain below.

LPSA is inspired by Latent Semantic Analysis (LSA), a theory of representation that explains how meanings of words can be learned from exposure to large amounts of experience. LSA is applied to understand language comprehension phenomena, and is much used in contemporary cognitive science. LPSA is a generalization of LSA where the environment does

not need to be text, and where the mechanism that LSA proposes to be working in the acquisition of meaning is considered a general mechanism for learning other complex activities.

One central idea of LSA and LPSA is that we can infer a representation that is optimal for doing generalization and similarity judgments in a particular environment using only information derived from co-occurrences between tokens and contexts. LSA and LPSA are spatial theories of representation based on the assumptions and concepts of Shepard (1987). That is, the proximal stimulus is supposed to be represented as a point in a multidimensional space, where all other past experiences can be represented as well. The space is created in such a way that it represents the similarities between objects. Thus, two objects that are similar tend to occupy areas close together in the mental space.

The remainder of this section is divided into two parts. First, we describe LSA as it was originally conceived by Landauer and Dumais (1997). Then we introduce LPSA and present evidence for the claim that it can be used as an alternative methodology to work with complex, dynamic tasks.

It is important to note that LSA and LPSA are both theories of cognition and technologies. In this paper, we are concerned with presenting LPSA as a technology that can help experimental cognitive scientists and practitioners to deal with complex task domains.

LSA

The focus of LSA is to explain the acquisition and representation of meaning. LSA uses a multidimensional space as representation and dimensionality reduction as a learning mechanism. LSA analyzes huge amounts of written text, capitalizing on the statistical structure of the occurrences of words in contexts. LSA induces representations of the meaning of words by analyzing the relation between words and passages in large bodies of representative 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 other areas (Landauer & Dumais, 1997).

LSA 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. As Kintsch (2001) put it, why did an author choose a particular word in a specific place instead of some other alternative? 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 surprising degree of success in areas like automatic essay grading (Foltz, Laham, & Landauer, 1999) and automatic tutoring to improve summarization skills in children (E. Kintsch et al., 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).

LSA is part of the empiricist tradition in that it assumes that the mind does not begin with detailed sets of principles and procedures (e.g., theories of morphological structure, case marking, etc.). Instead, it is assumed that a human brain begins with general operations for

association, pattern recognition, and generalization that, when applied to the rich structure of the environment, are sufficient to learn natural language.

LSA has been thoroughly described in the literature (e.g., Kintsch, 1998; Landauer & Dumais, 1997; Landauer, Foltz, & Laham, 1998), and it is currently widely used in different fields of cognitive science. Here we follow a very succinct explanation adapted and expanded from Steinhart (2000) and Landauer (unpublished). The best way to understand the LSA induction mechanism is through an example, on text passages and words, and the best small-scale example is described in Landauer, Foltz, and Laham (1998). The interested reader is recommended to consult these references for a better understanding of the technique.

From a pragmatic standpoint, LSA is a machine learning technique that offers the mechanisms to measure the semantic distance between two “passages” (where a passage is simply defined as a collection of one or more words) in a "semantic space," which is created from the input text. The space is generated by first constructing a matrix whose cell entries are the number of times a given word appears in a given passage. As an example, consider the passage “I have one sister and one brother.” If row i corresponds to the aforementioned passage and column j corresponds to the word “one”, then the value in cell ij would be 2 because “one” appears in the passage twice. The other words appear only once, so the values in the other cells in row i would be 1.

Real LSA spaces created this way are very big. For example, the “general knowledge” space TASA is representative of the size estimated for the cumulative reading of a college freshman. Since a passage contains only a small fraction of the number of unique words that make up the space, most of the entries in the matrix will be zeroes, and it is therefore a sparse matrix.

Once the matrix has been filled in, it is subjected to a mathematical technique called Singular Value Decomposition (SVD), which is similar in effect to training a neural network. This procedure constructs a semantic space that contains considerably fewer dimensions than the original matrix (typically 200-400). Each original word and each original passage is represented as a vector in this space.

The space can be viewed in an instructive but oversimplified way: Passages with similar semantic content are located near one another in the space, and words that have occurred in similar passages are also located near one another in the space. This view uncovers one of the benefits of dimension reduction—words that did not co-occur, but occurred in similar contexts (e.g., “doctor” and “physician”), will be grouped together as well. Because the words and passages in the space are arranged according to their semantic content, it is termed a "semantic space."

Once the space has been created, it can be used to compare novel passages, or those contained within the input set. The passages to be compared are represented as vectors (either already existing within the space for passages that were in the input set, or generated “on the fly” for novel passages by simply summing the vectors of the words contained in the passage), and the relation between these two vectors can be calculated. Typically, what is measured is the cosine between the vectors, which yields, in effect, the semantic distance between two words or passages. That is, the higher the cosine, the more related two items are. The interested reader can use LSA to compare passages or words by using the site http://lsa.colorado.edu.

LPSA

We saw that LSA offers a way of assessing semantic similarity between any two samples of text in an automatic, unsupervised way. Likewise, LPSA can evaluate the similarity between two samples of behavior in a work domain and produce a numeric result that corresponds quite well with human intuitions.

LPSA uses actions or states –tokens- instead of words, and samples of logged behavior of humans interacting with a system instead of samples of text. LPSA uses a whole trial as context. This definition of context is not arbitrary. In complex, dynamic tasks, there are long-term dependencies between tokens. That is, the consequence of a particular action might be visible only after a long time elapses. The selection of a working definition of context has to reflect this fact. A trial is the longest possible dependency between two states or actions: the worst case scenario is that the consequence of the very first action in a trial shows up at the end of it. Defining context as performance fragments smaller than the whole trial entails a risk: if some parts of the log are missing, causal chains may be broken and some important information may be lost.

Allowing for these differences, the procedure is parallel to the one described for LSA; the experimenter creates a matrix of types by trials. The matrix will be very sparse since most types appear in only a few trials. A schema of the matrix can be seen in *Figure 1*. This matrix is decomposed using SVD, and recomposed by multiplying the 3 resulting matrices, but using only a limited number of dimensions. In practice, the dimensionality that best captures human judgments is between 100 and 500. The resulting space can be considered a formal equivalent to the representation that people form when they interact with the task for an equivalent period of time. Since the corpus may contain performances from many different individuals, the resulting

space is an abstraction over subjects; that is, it does not represent a particular person, but an ”average” user of the system.

Any new slice of performance can be represented as a vector in this space and compared with previous activities. Performance from different individuals can be compared, as well as performances from the same individual on different occasions.

LPSA is flexible in terms of possible analysis: it enables both molecular analysis (e.g., at the level of actions or individual decisions) and molar analysis (e.g., several years of practice considered at the same time), as is shown in the applications described in this section.

If the task requires it, the cosines between LPSA vectors can be used as building blocks to construct theories that assume more high-level cognition, not only similarity-based operations. That is, the modeler can propose several layers of added assumptions (e.g., rule-based activity, mental models, etc.) using the basic LPSA space as starting point.

It is important to notice that the procedure is completely automatic, requesting the minimum use possible of the experimenter’s intuition to build a representation of the task. Our proposal is that the intuition of the experimenter should move from being the main explanatory factor (Kieras & Meyer, 2000) to being used only to resolve impasses in the automated process of deriving inferences from naturalistic samples of behavior.

Even though the description of the constraints in the environment is a key component in many theories of task analysis and work domain analysis (Vicente, 1999), there is no proposed method for inferring these constraints from the statistical regularities shown. LPSA is theoretically justified and part of an ample family of theories of knowledge representation in psychology (Chater & Vitanyi, 2003; Shepard, 1987; Shepard, 1994; Tenenbaum & Griffiths, 2001).To prove the theoretical power of the LPSA framework, we applied it to very different

complex tasks: A psychologically realistic theory should be able to explain very different domains without having to change its assumptions. To date, LPSA has been tested in three domains.

(1) LPSA was tested against human judgments of similarity (Quesada et al., submitted-b) emitted after watching a replay of a complex spatial task: Firechief. Firechief is a spatial microworld where participants control helicopters and trucks to extinguish a fire that spreads in a forest (see next section for a more detailed description of Firechief). Participants watched a randomly ordered series of trials, in a different order for each participant. The trials were paired and selected to sample different points from the LPSA similarity continuum (pairs A, B, C, D, E, F, G, with cosines 0.75, 0.90, 0.53, 0.60, 0.12 and 0.06 respectively)[2]. A high value for Cronbach's alpha indicated that the different human judges agreed and that the average of the human ratings had very high internal consistency. The comparisons with human judgments validate LPSA similarities very well: the correlation was .94.

(2) LPSA was tested on its ability to predict future states in Duress (Quesada, Kintsch, & Gomez, submitted-a). Duress is a thermodynamic microworld that simulates hydraulic process control, designed to be representative of industrial process control systems. It consists of two redundant feedwater streams that can be configured to supply either, both, or neither of two reservoirs. We used the data from a six-month-long, six-participants experiment reported in Christoffersen, Hunter, and Vicente (1996; 1997; 1998) to generate a "simulated expert" with three years of experience with the system. Given the first three quarters of any trial, our system could predict the last quarter with an average accuracy of .8. When the system was given an experience of only six months the accuracy of predictions fell to less than .3. If the system was trained with three years of practice but in an environment with no constraints (that is, not

governed by rules of conservation of mass and energy), the predictions were, as one might expect, not useful and were comparable to those of a human novice level. LPSA's explanation for these results was able to generate predictions that either process theories or product theories of expertise could explain; both have never been explained by a single computational theory.

(3) LPSA was tested against a realistic domain: a High-Fidelity Flying simulator, used to develop a landing technique automatic assessment system based on LPSA. Quesada, Kintsch, and Gomez (2003a; submitted-b) collected 400 landings where the landing conditions were manipulated systematically, and created an LPSA vector space with them. Two instructors evaluated the landing, one of them sitting in the copilot seat, and the other one watching plots of relevant variables in real time (complete and reduced information experts respectively). The model was trained with the variables that the reduced-information expert used in his plots to evaluate the landing technique. Then the nearest neighbors of any new landing were used to generate automatic ratings. The ratings that the model emitted agreed with both humans as much as the two human graders agreed with each other.

In summary, LPSA is an automatic procedure to represent task constraints using logs of naturalistic activity in very complex environments. LPSA proposes a common method for describing virtually any task domain that can be logged. Only more research and application can determine what (if any) environments are not appropriately described by this formalism. LPSA, unlike many cognitive task analysis approaches, is independent of both the experimenter and the task. That is, two experimenters generating a representation for the same task should come up with the same LPSA space. Therefore, LPSA spaces can be shared and some corpora can be used as reference points for a research community. For example, the longitudinal experiment of Christoffersen, Hunter, and Vicente (1996; 1997; 1998) can be considered already a classical

reference corpus that should be used by competing models trying to explain the same phenomena.

An example: Firechief and cognitive flexibility

LPSA has been applied successfully also as a technique, that is, a method to assess similarity that can be used in different applications, and as a dependent variable in experimental designs. In these situations, the LPSA-simulated average human is used not as a participant itself, but as a way to classify real participants into groups and evaluate changes in participant behavior over time, and as a general reference and measurement tool.

The Cañas et al. (2003) study on cognitive flexibility illustrates how LPSA can be used to bypass the main problems that we discussed above. The purpose of this experiment was to test whether people change their strategies when the environment changes. The experimenters trained participants in constant conditions in the microworld Firechief, and once the participants developed a strategy, the experimenters introduced a change in the environment. Their results showed that performance was affected by changes that were relevant to certain strategies, but not others. They conclude that to explain behavior and adaptation in complex dynamic tasks one should focus on the interaction between cognitive processes and environmental conditions, because neither factor alone is sufficient.

Cañas and collaborators developed a method for describing performance and analyzing "strategies" in Firechief (Cañas et al., 2003; Quesada, Cañas, & Antoli, 2000). First, they designed a set of very simple, simulated theoretical strategies using a production system. The strategies were coded as matrices containing transitions between two contiguous actions as the unit of analysis. These strategies were orthogonal, and were used as factors in a regression analysis applied to every trial to describe performance.

Cañas et al.'s measures imposed a priori, theoretically driven assumptions on their analyses. These assumptions are reasonable and well motivated, but they may nevertheless bias the outcome of the analyses in unknown ways. (1) They used "transition between two actions" as their unit of analysis. However, they could have also used three actions, or more, and this change could have had important consequences. (2) To avoid the empty space phenomenon,[3] they used only part of the information from the log files (type of action); although type of action probably was the most important part, much potentially useful data was lost in this process. (3) They used a multiple regression approach to compare empirical "strategies" and theoretical ones (factors). This procedure relies on correlation, and this similarity measure has several flaws (see Jones & Furnas, 1987 for a comparison of similarity measures). (4) To interpret groups and for other reasons, they used a set of simple theoretical (simulated) strategies. The designed strategies were orthogonal in the sense that their matrices did not correlate. However, the selection of theoretical strategies can strongly determine the results: How many of them do we need? Which of them? Even strategies created using task analysis and following the simplicity criterion impose an a priori, possibly unwarranted theoretical structure on the data. (5) They used a k-means cluster analysis procedure over dichotomous values of significance (0 or 1) of beta values in the multiple regression analysis, further imposing additional structure on their data.

Our solution uses the LSPA cosines as a continuous dependent variable for assessing the change in control behavior of people over time, and shows how to use LPSA as a dependent measure in experimental designs. LPSA cosines are used in two ways: (1) as a measure of strategy change over time (this is a very sensitive process measure), and (2) for grouping participants according to performance before changes.

Below, the LPSA-derived method is applied to the same dataset used in Cañas et al. (2003) to reproduce their effects without the methodological issues raised. We present a coherence measure to assess how participants change their control activities to cope with changes in the simulated environment. Coherence is defined as the similarity of the current trial to a window including several previous trials. This measure resembles Hajdukiewicz and Vicente's (1999; 2000) Within-trial Trajectory Deviation (WTD) method. WTD is a method performance in Duress trials. Since Duress consists mainly of continuous variables, changes can be calculated by simply subtracting the values of successive variables. Trajectories are plots of the differences between vectors representing the states of the system over time. The bigger the area under the curve, the more different are the trials compared. The basic idea is to compare the trajectory of a target trial to the mean trajectory of a set of previous trials. Hajdukiewicz and Vicente did that by means of a sliding window; that is, each trial was compared with a small window of previous trials. This measure indicates the amount of change in the performance for each trial. In our method, the comparisons between the window and the current trial employ LPSA cosines. This measure is inspired by the work of Foltz, Kintsch, and Landauer (1998), who were able to predict text comprehension using the average LSA cosine between vectors corresponding to contiguous sentences. They calculated the vector in the LPSA space for each sentence, and then the cosine between successive sentences. The average cosine for each text was an indicator of how coherent the text was. Texts with higher coherence were more easily understood. In one of their experiments, they used a data set from McNamara, E. Kintsch, Songer, and Kintsch (1996) where, in some conditions, word overlap between sentences was minimal. Even in those circumstances, LSA performed well at capturing the coherence of the text. This feature of LSA, perpetuated in LPSA, makes it very interesting for microworld

performance comparisons, since the same actions are rarely used, even when the intentions of the participants are quite similar in two trials.

Method

Participants. 81 participants were asked to play 22 trials each on Firechief (Omodei & Wearing, 1995) in two nonconsecutive sessions (not on the same day, though no more than four days apart), the first of an hour and a half and the second approximately an hour. In the first session they played 10 experimental trials and in the second one 12. Each experimental trial lasted 260 seconds. During the first 30 minutes of the first session the experimenter explained the task and ran three practice trials to train participants in the commands and the characteristics of the task. Only trials 13-20 were considered in these experiments (four trials before the change, and four trials after it).

Apparatus. In Firechief (Omodei & Wearing, 1993, 1995), participants are confronted with a simulation of a forest through which a fire is spreading. Their 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), which can be controlled by mouse movements and key presses. The different cells have different ratings of flammability and value: houses are more valuable than tree cells, for example. The participant's mission is to save as much forest as possible, to preserve the most valuable cells, and to prevent the trucks from being burned. Helicopters move faster and drop more water than trucks, but the latter can make control fires. Trucks are unable to extinguish certain fires, and they will be destroyed if they are sent to them. The fire is more intense and spreads faster depending on the wind direction. Wind direction and strength are indicated in the top-right corner of the interface. The wind changes systematically in some conditions (see, Cañas et al., 2003, for a description of the conditions). Participants can see a

window with their overall performance score at the end of a trial, which is calculated by adding every safe cell and subtracting the value of the burnt trucks. It is possible to experimentally control features of the system, and to prepare experimental situations for testing a wide variety of hypotheses.

There are three commands that can be used to control the movement and functions of the appliances: (1) drop water on the current landscape; (2) start a control fire (trucks only) on the current landscape segment, and (3) move an appliance to a specified landscape segment. Commands are given using a ”drag-and-drop” method by first selecting the desired vehicle (by moving the mouse cursor into the landscape segment containing it) and then pressing a key on the keyboard. At the end of each trial, the program saves the command sequence that the participant issued in that trial. These log files were used to train LPSA.

Procedure. Participants were assigned randomly to two groups with different environmental changes, which were designed to affect systematically two prototypical "strategies" (e.g., patterns of actions) that were found in previous experiments (Quesada et al., 2000). The first 16 trials were identical for the two groups. On these trials, conditions were held constant in terms of fire distribution, wind direction, appliance characteristics, etc.

The last six trials introduced one of two system manipulations: (1) The wind direction change group experienced a progressive, east-to-west wind change. This kind of manipulation is known to affect mainly participants who rely on control fires, which have to be relocated. (2) The appliance efficiency change group experienced a drastic reduction in the extinction power of both helicopters and trucks. Fires of half the size were now impossible to extinguish by dropping water on them. This kind of manipulation was supposed to interfere with the performance of

participants who rely on the drop-water command, since the fires are now more difficult to stop this way.

The remaining features of the systems were kept exactly the same as in the 16 previous trials. Participants did not know beforehand that a manipulation was going to be introduced. Since microworld settings are, in general, difficult to describe in the procedure section of a paper in a way that enables perfect replication, the parameter files used are available upon request.

LPSA corpus creation. To create an LPSA space for present purposes, we used as the corpus data from experiments 1 and 2 described in Quesada et al. (2000), plus data from the experiment described in Cañas et al. (2003). The conditions and changes in these experiments were identical to the wind change condition described above. Actions were coded by joining the information contained in one line of the log files. It is important to note that this procedure does not require the experimenter to select part of the information (as in the procedure described in Quesada et al., 2000), but can cope with the whole action description (all variables defining a state). However, since LPSA must be trained on a set of variables that should resemble closely the information that participants use to perform in the task, the experimenter's intuition has to identify the information that participants most probably do not use. In this case we dropped from the logs the information on (1) appliance number and (2) departure coordinates. This decision is motivated by some empirical findings in spatial attention when applied to multiple moving targets, most of them reported by Pylyshyn (e.g., Pylyshyn, 1994, 2001; Scholl, Pylyshyn, & Feldman, 2001). In Pylyshyn's experiments, participants had to keep track of up to four items (targets) that moved randomly in a field of eight or more identical items (distractors). After all the items stopped moving, participants pointed out which ones were the targets. The assumption was that observers who had tracked the targets correctly also had kept track of their individual

identities. Thus they should be able not only to identify the objects as members of the target set, but also to recall other identifying information initially associated with them such as names, colors, or place of origin. This experimental situation very much resembles the one that our participants experienced in the human judgments experiment. Pylyshyn found that people can track the items, but not recall their identities; in our case, people should not be able to say whether one truck is truck 1, originally in cell (11, 9), or truck 2, which started in cell (4, 11).

This is an example of how experimenter expertise is used to guide the modeling process only when the model cannot make an informed decision.

These variables were removed from the log files that LPSA used as input. Thus, instead of 360199 actions in 3441 trials, the reduced corpus had 3627 different actions in 3441 trials. After performing singular value decomposition on the actions by trials matrix (SVD), we kept 319 dimensions. Manipulating the dimensionality between 100 and 1500 generated similar results, so we will only report the model that used around 300 dimensions.

Grouping participants according to performance before changes. A third independent variable was generated a posteriori; our basic assumption was that participants with different ways of controlling the system get different results and are not completely comparable in a task with that many degrees of freedom. To alleviate this problem, we clustered participants according to their most commonly used actions.

To do so, we used their log files from trials 1-16. In past studies, the "strategies" were hand-coded and implemented as a production system that was exposed to the same environmental conditions (Cañas et al., 2003; Quesada et al., 2000). The log file produced by the simulated strategy was compared to the ones from human participants. Two strategies were designed: (1) a drop-water strategy, based on movements to the coordinates of the fires, and

actions of dropping water over them, and (2) a control-fire strategy, with movements to areas close to the fire front and actions directed to set up control fires to prevent the expansion of the fire. In this study, hand-coded strategies have been eliminated. Instead, we used a vector obtained by averaging all the relevant actions that appeared in the corpus. A vector for the control-fire strategy was created by averaging all 411 actions containing control-fire, and a vector for the drop-water strategy was created by averaging the 449 actions containing drop-water. These two vectors were unrelated to each other, and their cosine was -0.17. Each subject's vector was then compared to these two pure strategy vectors. Of course, real subjects do more than drop water or set control fires, so the average cosines between the subject vectors and the strategy vectors are very low. Nevertheless, when these cosines were clustered (via K-means), three distinct and very different clusters emerged, as shown in Figure 2. In our analysis, we impose structure after asking for 3 clusters. This is just to mimic the decision made in the original paper (Cañas et al., 2003).

Thirty-one participants who preferred the dropping water actions compose Cluster 1. Cluster 3 represents those 19 participants who preferred to make control fires instead. Cluster 2 is a hybrid, where people performed both actions in roughly equal proportions (33 participants).

Design: There were 3 independent variables: (1) environmental change type, with two levels (wind change and appliance efficiency change), manipulated between groups, (2) cluster, with three levels (Cluster 1, 2 and 3), manipulated between groups, and (3) before-after change, with two levels (four trials immediately before change averaged, and four trials during the change averaged), manipulated within subject.

There were two dependent variables, one product measure and one process measure: (1) overall performance defined as the sum of all cells that remained safe, subtracting the value of all

burnt trucks at the end of the trial, and expressed as a proportion of the total area, and (2) coherence, defined as the similarity of the current trial to a window including several previous trials. The more important the change in how the participant controls the system, the lower the coherence becomes. The window size was four trials. Different numbers of window trials (e.g. 5 or 3) did not change the results significantly. This window size is similar (proportionally) to the one used in Hajdukiewicz and Vicente (1999), who used a window of 40 trials for a total of 220 trials. For each trial, we averaged the vectors for the four previous trials and then compared the average to the vector of the current trial. The cosine between these two vectors indicated how consistent a participant was in her way of controlling the system; using this coherence measure we were able to detect when a participant had changed her approach to the task, because the cosine would drop noticeably when a change was introduced. Since we wanted to compare the change trials (17, 18, 19, 20) to the ones before change, the sliding window was stopped before the change was introduced: the coherence measure was calculated by comparing the average of 13, 14, 15 and 16 to trial 17, the average of 13, 14, 15 and 16, to trial 18, etc.

Predictions. We predicted a drop in performance when the new environmental conditions were introduced. The drop in performance should be bigger for those participants whose log files did not show any change in their pattern of action, that is, those whose actions under the new conditions still resembled their actions on the previous trials. We also predicted that those participants who did adapt to the changes would exhibit a coherence drop.

Results

Two mixed ANOVAs (environmental change type, cluster, EG; before–after change, IS) were performed on the two dependent variables. For overall performance, the significant main effects were environmental change type ($F[1.77] = 9.00$, $MSE = .033$, $p<0.01$) and before–after

change ($F[1, 77]= 27.06$, $MSE = .004$, $p < 0.01$). The interaction between these two factors was also significant [$F(1,77) = 11.674$, $MSE = .004$, $p < 0.01$]. Figure 3 (b) shows that, while participants exposed to the appliance efficiency change did worse after change was introduced, participants exposed to the wind change condition adapted well and suffered no significant performance drops.

Note that the results for the coherence measure show a reverse, mirrored pattern: The same main effects ($F[1,77] = 11.23$, $MSE = .0081$, $p < 0.01$) ($F[1,77] = 61.34$, $MSE = .0069$, $p < 0.01$) and the second order interaction ($F[1,77] = 22.89$, $MSE = .1596$, $p < 0.01$) are significant. The group with high coherence after the change is the one that shows a big decrease in performance, whereas participants who did not repeat the same actions as before the change (depicted as a fall in coherence) maintained a good overall performance (see Figure 3[a] and Figure 3[b]).

Although the groups seemed to start at different levels of performance, the difference in performance before changes (Figure 3b, .75 vs. .80) was not significant ($t[80] = -1.1591$, $p > 0.20$), and their coherence was approximately the same ($t[80] = -0.9873$, $p > 0.30$).

Third order interactions. Since we grouped participants according to their way of controlling the system (see cluster analysis results), we could check whether different clusters behaved differently when the changes were introduced. The third order interactions answer this question. For overall performance, the third order interaction was significant ($F[2,77] = 10.21$, $MSE = .0041$, $p < 0.01$): participants assigned to different clusters and different environmental changes were affected by changes in a different way. This interaction was not significant when the analysis was performed using the coherence measure as the dependent variable ($F[2,77] = .8790$, $MSE = . 0061$, $p < 0.42$). However, a closer analysis reveals an interesting pattern of

results. Figure 4 (c) shows that participants located in Cluster 3 who experienced the change in appliance efficiency were very much affected in their overall performance, ($t[10] = 4.3458$, $p < 0.01$). Interestingly enough, Figure 5(c) shows a very stubborn behavior, with no changes in coherence ($t[10] = 0.4352$, $p > 0.60$).

Figure 5(a) shows that some participants who experienced the change in appliance efficiency were able to adapt to the change. The comparison before–after for Cluster 1 participants in appliance efficiency change showed a significant drop in coherence ($t[10] = 1.9495$, $p < 0.05$), and no impoverished overall performance (Figure 4a) in change situations ($t[10] = 1.7652$, $p > 0.10$).

Discussion

Our analysis replicates and extends the results obtained in Quesada et al. (2000) and Cañas et al. (2003). These previous studies did not compare "strategies" before and after the change was introduced, although this comparison would have been relevant to their conclusions. The main reason is that their methodology provided no continuous measure of similarity between two trials. Their procedure depended on strategies defined a priori, with the similarity to these strategies coded all or nothing, using only significant correlations between matrices. Since this measure did not provide a continuous way of assessing similarity, it would not have been very meaningful to compare pairs of trials, or even groups of trials (averages) to individual trials. The coherence measure introduced in this paper has proved to be sensitive enough to capture differences in participants' performance before and after the change is introduced.

It is informative to compare what LPSA can do with the limitations of the other methods cited for Firechief and Duress analyses. (1) The problem of using transitions between actions as the unit of analysis is that it assumes that actions depend only on the previous (and next) action.

This is a strong assumption in a complex system like the ones generated in microworlds. Since LPSA uses dependencies among all actions in all contexts to infer the relationships among them, it does not rely on a unit of analysis selected a priori to represent dependency. (2) The combinatorial explosion that would render the "transition between actions" procedure unusable if more variables were chosen does not affect LPSA, because the SVD reduces the dimensionality of the space, making the "empty space phenomenon" less of an issue. In fact, if one tries to do the same analysis of transitions with all the variables included (ignoring the simplifying assumptions used in Cañas et al) the results do not correlate with human judgments ($r = -.39$). (3) The multiple regression analysis that relies on correlation is replaced with the cosine measure, which has been shown to better capture similarity relationships (Jones & Furnas, 1987). (4) The selection problem in designing a set of a priori strategies is alleviated, although not eliminated in this particular replication. An abstract vector that represents the experimenter's hypothesis replaces the hand-coded, rule-based implementation of "strategies." However, this part of the procedure was kept for replication purposes, and is not a necessary part of the LPSA method. Future analysis will not necessarily rely on it. (5) The dichotomy of the significant versus non-significant correlation is replaced by the cosine measure, a continuous one.

Through the use of LPSA we could cluster participants without defining strategies by implementing production systems. This is an important advantage, for two main reasons. (1) It enables experimenters to work with systems that they do not necessarily know very well. In real applications (e.g., flying simulators or nuclear power plant control rooms) the knowledge to implement a production system that performs theoretical strategies is, most of the time, out of reach for the experimenter; the bottom-up approach described here (abstract representation of the actions/states that represent a particular control behavior) can be implemented automatically and

overcomes the expertise problem. (2) For the same reason, the LPSA procedure described here enables researchers to work with bigger, more complicated systems.

Our results throw some light on the use of spatial information by the participants in this experiment. The only group that actually improved performance after change was the one formed by participants in Cluster 3, subjected to the manipulation wind change (Figure 3c, dashed line). Interestingly enough, this group initially preferred to use control-fires (Figure 2) it adapted to the change in wind direction not by abandoning this strategy and shifting to a drop-water strategy (the control-fire vector is not significantly different than before the changes, $t(7) = 1.97$, $p < 0.09$), but by appropriately changing the spatial location of the control fires. Since LPSA "knows" or "pays attention to" the spatial locations (coordinates in the log files), our analysis was able to reflect this change in strategy.

General discussion and conclusions

We have presented some criticisms of the way theories about task representation are proposed. Specifically, researchers have pursued task decomposition even though there are no guarantees that it is the right way to approach complex tasks. In some other areas of cognitive science both holistic and decomposition approaches are used; task theories should be trying both methods as well. Experimenter introspection is the main tool used both to perform the task decomposition and to generate the “strategies” that people use, and this trust in introspection and intuition may be hampering the development of integrative theories.

We addressed these criticisms by proposing LPSA as an alternative method of representing tasks and environment. If the environment to model is highly structured in a way that is known beforehand, it would be better to use a method that allowed for structured representations; however, even though LPSA is a metric space that does not allow for structured

representations, it can capture a surprising amount of variability. There might be other formalisms better suited to a particular characteristic of the domain. For example, if the experimenter has reasons to believe that she may benefit from a procedure that generates structured representations, for example tree-based ones, she may pick an inference method that produces that particular structure. However, it seems safe to assume that the "lowest common denominator" is the metric space that the current implementation of LPSA uses. LPSA has been validated with Firechief, Duress, and a high-fidelity flying simulator. Any alternative method should be tested against these datasets to make sure that it can account for them and for some other effects and observations that are out of reach for LPSA.

This example used LPSA cosines in a hypothesis testing framework (as a dependent variable). However, one can envision LPSA experiments for hypothesis generation instead of hypothesis testing. For example, one can create an LPSA space leaving out the variables that, according to one's hypotheses, are not used by humans. If this is true, the space would still predict human similarity judgments as well as before.

We do not want to tie our approach to the impossibility of decomposing complex activities. If someone in the future demonstrates that any complex task can be decomposed into smaller units, our framework would still be applicable. LPSA does not assume decomposability , but does not depend on tasks being non-decomposable. However, we do want to show that there are valid alternatives to the intuitive decomposition of tasks.

The last criticism we raised at the outset was related to the importance of strategies. Although we prefer not to use the term "strategies" because it is very difficult to define, one can think of them as hyper-volumes in the LPSA hyperspace. Similar solutions to the same problem (which could be accommodated under the tag of a "shared strategy") would be located in similar

regions of the space. The shape of this region can be very convoluted, describing a trajectory through the multidimensional space. The mathematical properties of these ”strategy volumes” could be studied formally, if the experimenter needed to do so.

A related problem is the way theories are derived once an experiment is finished in a particular, specialized context. Xiao and Vicente (2000) pointed out the epistemological dangers in leaping from field work and experimental data to generalizable theories, and presented a framework to reach the much-wanted general theory in concentric layers of abstraction, in a way similar to their Abstraction Hierarchy. However, the way each layer is created is simply an exercise of experimenter intuition. Xiao and Vicente (2000) showed concern about the reliability of this abstraction process. We share their concern, since it is not guided by any systematic formalism.

This danger is accentuated by the lack of systematic abstraction procedures for going from one layer of data to another. An ideal solution should propose a formal method to generate new abstraction layers using the experimenter’s knowledge only as an aid to solve impasses where the evidence is not enough to make an informed decision.

LPSA can be run at different levels of abstraction; in fact, this is what Quesada et al. (2003b) did for the analysis of Duress performance, using the Abstraction Hierarchy presented by Vicente (1991) for that task. However, the Abstraction Hierarchy is not part of LPSA as a theory, because it would be against the general principle of using the minimum amount of experimenter’s a priori knowledge to define the representation of the task. There are no mechanisms in the current implementation of LPSA to generate the empirically induced equivalent to the Abstraction Hierarchy. A promising future line of work will be to generate representations at different levels of abstraction automatically.

In summary, complex, dynamic systems such as the tasks that interest the human factor community still show strong statistical regularities. The most common approach to creating theories about how people derive representations for these regularities and use them in their daily interaction with the systems has been to use the experimenter's common sense instead of more formal models of inference. This has led to a fragmented theory that poses serious epistemological problems and is difficult to generalize. Our proposal is to leave abstraction to formal inference procedures as much as possible, and represent the constraints of the system in a bottom-up approach assisted by the experimenter's intuition only at particular moments. In this sense, we would depict intuition as an assistant, not a director, in the process of generating task representations. Nowadays, there are abundant datasets, computing power, and efficient algorithms to make this idea possible. LPSA exemplifies how these ideas could be implemented. Existing microworld experiments could be used as datasets, and the SVD as a method to generate metric representational examples. But the framework presented is not limited to these choices: we expect it to work with real activity samples logged from any work domain of interest. We also expect different algorithms for inference to be better suited for future applications. In general, we believe that this framework could be important to the advance of theories of the environment (task), an area of cognitive science that seems to have fallen behind compared to the advances we have made in theories about cognitive processes and representations.

References

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, New Jersey: Lawrence Erlbaum associates.

Bainbridge, L. (1999). Verbal reports as evidence of the process operator's knowledge. *International Journal of Human-Computer studies, 51*, 213-238.

Biederman, I. (1987). Recognition-by-components: A theory of human understanding. *Psychological Review, 94*(2), 115-117.

Brunswik, E. (1956). *Perception and the representative design of psychological experiments*. Berkeley, CA: University of California Press.

Cañas, J. J., Quesada, J. F., Antolí, A., & Fajardo, I. (2003). Cognitive flexibility and adaptability to environmental changes in dynamic complex problem solving tasks. *Ergonomics, 46*(5), 482-501.

Chater, N., & Vitanyi, P. (2003). The generalized universal law of generalization. *Journal of Mathematical Psychology, 47*(3), 346-369.

Christoffersen, K., Hunter, C. N., & Vicente, K. J. (1996). A longitudinal study of the effects of ecological interface design on skill acquisition. *Human Factors, 38*, 523-541.

Christoffersen, K., Hunter, C. N., & Vicente, K. J. (1997). A longitudinal study of the effects of ecological interface design on fault management performance. *International Journal of Cognitive Ergonomics, 1*, 1-24.

Christoffersen, K., Hunter, C. N., & Vicente, K. J. (1998). A longitudinal study of the impact of ecological interface design on deep knowledge. *International Journal of human-Computer Studies, 48*(6), 729-762.

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.

Edelman, S. (1998). representation is representation of similarities. *Behavioral and Brain Sciences, 21*(4), 449-498.

Edelman, S. (1999). *representation and recognition in vision*. Cambridge, Massachusetts: MIT Press.

Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual Coherence with Latent Semantic Analysis. *Discourse Processes, 25*, 285-307.

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)*.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco, CA: W.J. Freeman and Company.

Gigerenzer, G., & Goldstein, D. G. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review, 103*(4), 650-669.

Gordon, A. S. (2003). *Strategy Representation: An Analysis of Planning Knowledge*: Lawrence Erlbaum Associates.

Hajdukiewicz, J. R., & Vicente, K. J. (1999). *A cognitive engineering approach for measuring adaptive behavior*: Cognitive Engineering Laboratory: University of Toronto (CEL).

Hajdukiewicz, K. R., & Vicente, K. J. (2000). Ecological interface design: adaptation to dynamic perturbations, *Proceedings of the fifth International Conference on Human interaction with complex systems* (pp. 69 - 73). Urbana-Champaign, IL: The Beckman Institute.

Hoffman, R. R., & Woods, D. D. (2000). Studying cognitive systems in context: preface to the special edition. *Human Factors, 42*(1), 1-7.

Jones, W. P., & Furnas, G. W. (1987). Pictures of relevance: A Geometric Analysis of similarity measures. *Journal of the American society for information science, 38*(6), 420 -442.

Juslin, P., Olsson, N., & Winman, A. (1996). Calibration and diagnosticity of confidence in eyewitness identification: Comments on what can be inferred from the low confidence-accuracy correlation. *Journal of Experimental Psychology-Learning Memory and Cognition, 22*(5), 1304-1316.

Kieras, D. E., & Meyer, D. E. (2000). The role of cognitive task analysis on the application of predictive models of human performance. In J. M. Schraagen & S. Chipman & V. J. Shalin (Eds.), *Cognitive task analysis* (pp. 237 - 261). Mahwah, New Jersey: Lawrence Erlbaum associates.

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

Kintsch, W. (1998). *Comprehension: a paradigm for cognition*: Cambridge University Press.

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

Landauer, T. K. (unpublished). Some remarks on consciousness by a somewhat maverick cognitive scientist.

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.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes, 25*, 259-284.

Lee, F. J., & Anderson, J. R. (2001). Does learning a complex task have to be complex? A study in learning decomposition. *Cognitive Psychology, 42*, 267-316.

McNamara, D., Kintsch, E., Butler Songer, N., & Kintsch, W. (1996). Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction, 14*(1), 1-43.

Newell, A. (1990). *The unified theories of cognition*: Harvard University Press.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Omodei, M. M., & Wearing, A. J. (1993). Fire Chief (Version 2.3): University of Melbourne.

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.

Payne, J. W., Bettman, J. R., & Johnson, E. J. (1993). *The adaptive decision maker*. New York, NY: Cambridge University Press.

Pylyshyn, Z. W. (1994). Some primitive mechanisms of spatial attention. *Cognition, 50*, 363-384.

Pylyshyn, Z. W. (2001). Tracking without track: Loss of item identity in multi-element tracking. *Meeting of the psychonomic society, (41ST annual meeting), 5*, 34.

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

Quesada, J. F., Kintsch, W., & Gomez, E. (2003a). Automatic Landing Technique Assessment using Latent Problem Solving Analysis, *25th Annual Conference of the Cognitive Science Society*.

Quesada, J. F., Kintsch, W., & Gomez, E. (2003b). Latent Problem Solving Analysis as an explanation of expertise effects in a complex, dynamic task, *Proceedings of the 25th Annual Conference of the Cognitive Science Society*. Boston, MA.

Quesada, J. F., Kintsch, W., & Gomez, E. (submitted-a). Expertise as the creation of multidimensional spaces. *Psychonomic Bulletin & Review*.

Quesada, J. F., Kintsch, W., & Gomez, E. (submitted-b). Latent Problem Solving Analysis (LPSA): A theory of representation in complex problem solving. *Cognitive Science*.

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.

Schank, R. C. (1972). Conceptual dependency: A theory of natural language processing. *Cognitive psychology, 3*, 552-631.

Scholl, B. J., Pylyshyn, Z. W., & Feldman, J. (2001). What is a visual object? Evidence from target merging in multiple object tracking. *Cognition, 80*(1-2), 159-177.

Schraagen, J. M., Chipman, S., & Shute, V. J. (2000). State-of-the-art review of cognitive task analysis techniques. In J. M. Schraagen & S. Chipman & V. L. VShalin (Eds.), *Cognitive task analysis* (pp. 467 - 487). Mahwah, New Jersey: Lawrence Erlbaum associates.

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

Shepard, R. N. (1994). Perceptual-cognitive universals as reflections of the world. *Psychonomic Bulletin & Review, Vol 1(1)*, 2-28.

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

Steinhart, D. (2000). *The LSA-based reading and writing tutor, Summary Street.* Unpublished doctoral dissertation, University of Colorado, Boulder.

Tenenbaum, J. B., & Griffiths, T. L. (2001). Generalization, similarity and Bayesian inference. *Behavioral & Brain Sciences, Vol 24(4)*, 629-640.

Vicente, K. J. (1991). *Supporting Knowledge-Based Behavior Through Ecological Interface Design.* University of Illinois at Urbana-Champaign.

Vicente, K. J. (1999). *Cognitive Work Analysis*. Mahwah, NY: LEA.

Xiao, Y., & Vicente, K. J. (2000). A framework for epistemological analysis in empirical (laboratory and field) studies. *Human Factors, 42*(1), 87-101.

Yu, X., Lau, E., Vicente, K. J., & Carter, M. V. (1998). Advancing performance measurement in cognitive engineering: The abstraction hierarchy as a framework for dynamical system analysis, *Proceedings of the human Factors and Ergonomics Society 42nd Annual meeting*. Santa Monica, CA: Human factors and Ergonomics Society.

Author Note

José Quesada, Institute of Cognitive Science, University of Colorado, Boulder, Campus Box 344, Boulder, CO 80309-0344,quesadaj@psych.colorado.edu.

The simulator and expert time was possible thanks to a grant supported by the European Community - Access to Research Infrastructure action of the Improving Human Potential Program under contract number HPRI-CT-1999-00105 with the National Aerospace Laboratory, NLR This research was also supported by Grant EIA – 0121201 from the National Science Foundation and by the grant BSO 2002-02166 from the Ministerio de Ciencia y tecnología, Dirección General de Investigación.

Our acknowledgements to Tom Landauer, Simon Dennis and Bill Oliver, for proposing interesting theoretical issues. We are grateful to Kim Vicente and John Hajdukiewicz for sharing experimental data and insightful discussions. We also thank Nancy Mann for a very thorough editing work.

Footnotes

1 The same example has also been used in Deerwester (1991) and Landauer & Dumais (1997).

2 This method was selected so the cosines were very distinctive and easy to compare. A subspace where a few items (instead of pairs) have very distinctive cosines could be found too, but this approach was used because the sample of stimuli was easier to obtain.

3 Due to the curse of dimensionality, matrices of transitions between actions are mostly zeros; since any pair of matrices to be compared are sharing a lot of zero cells, correlations are very high, but uninformative (artifactual).

Figure Captions

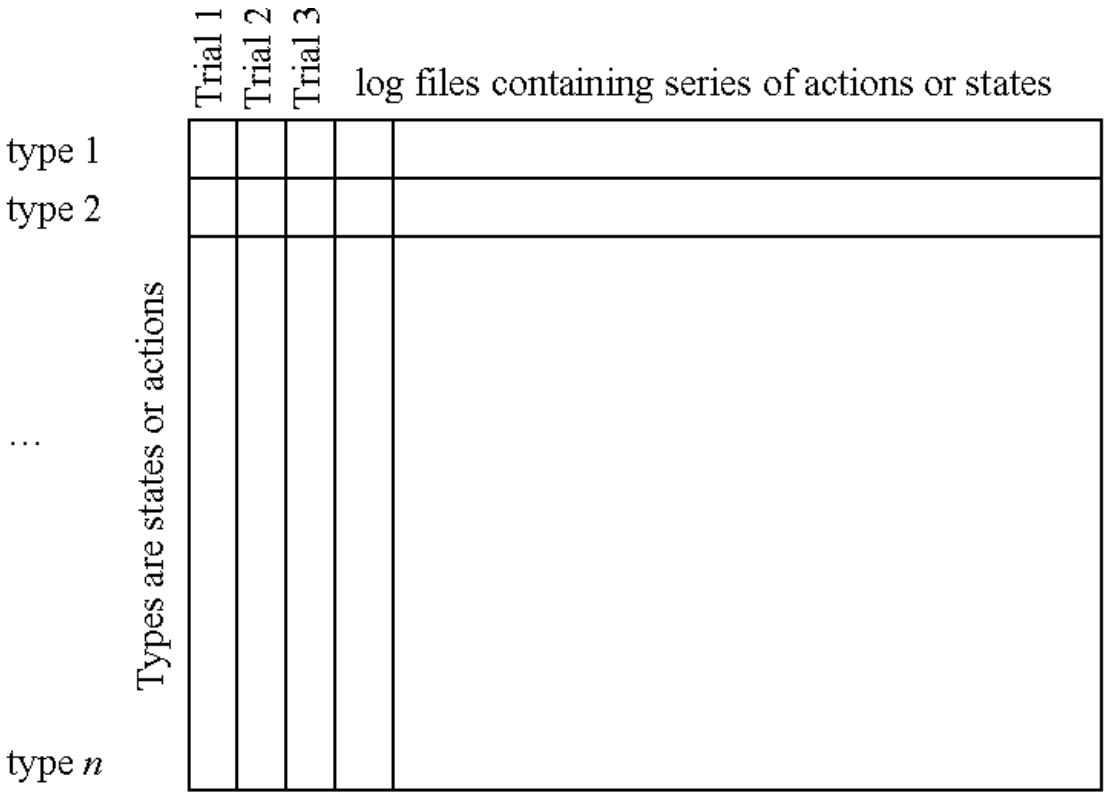
Figure 1: Matrix of types by contexts.

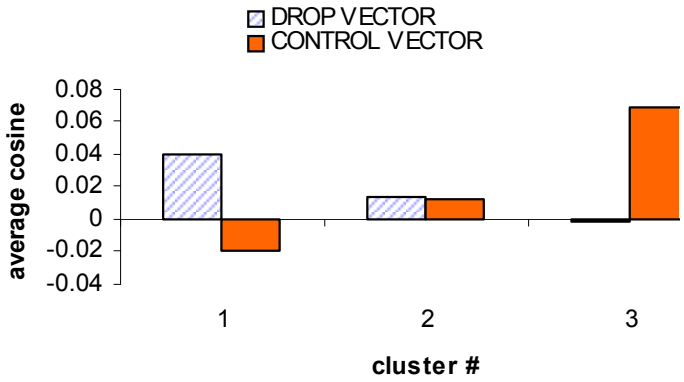
Figure 2 K-mean clusters using cosines to the drop and control strategy vectors.

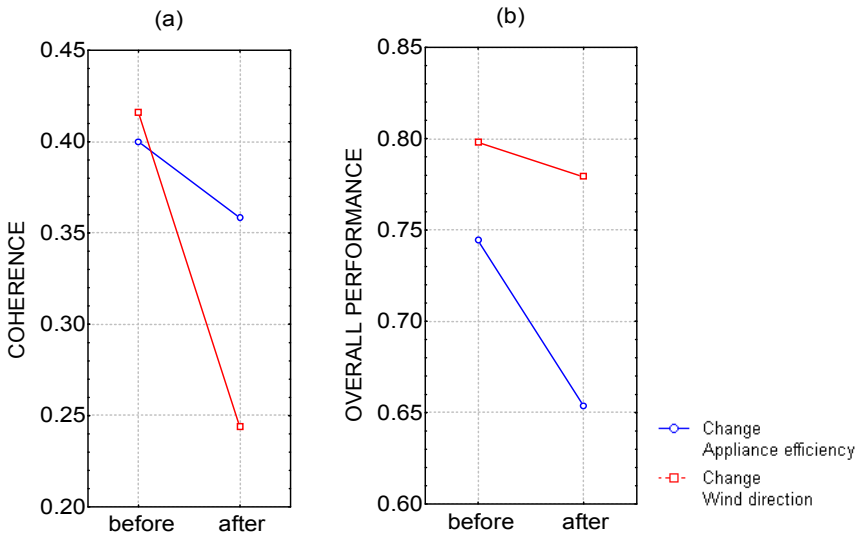
Figure 3 Interactions between environmental change type and before–after change, for two ANOVAs using Coherence and Overall Performance as dependent variables. "Before" stands for the average of 4 trials before change (13, 14, 15, 16), and "after" is the mean of trials that featured the change (17, 18, 19, 20).

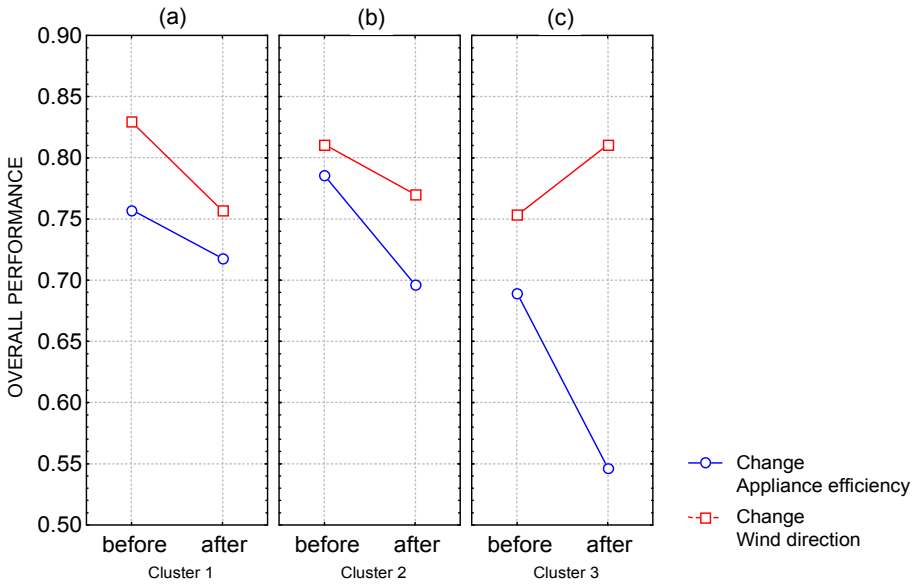
Figure 4 Interaction between environmental change type, before–after change, and cluster group using Overall Performance as the dependent variable

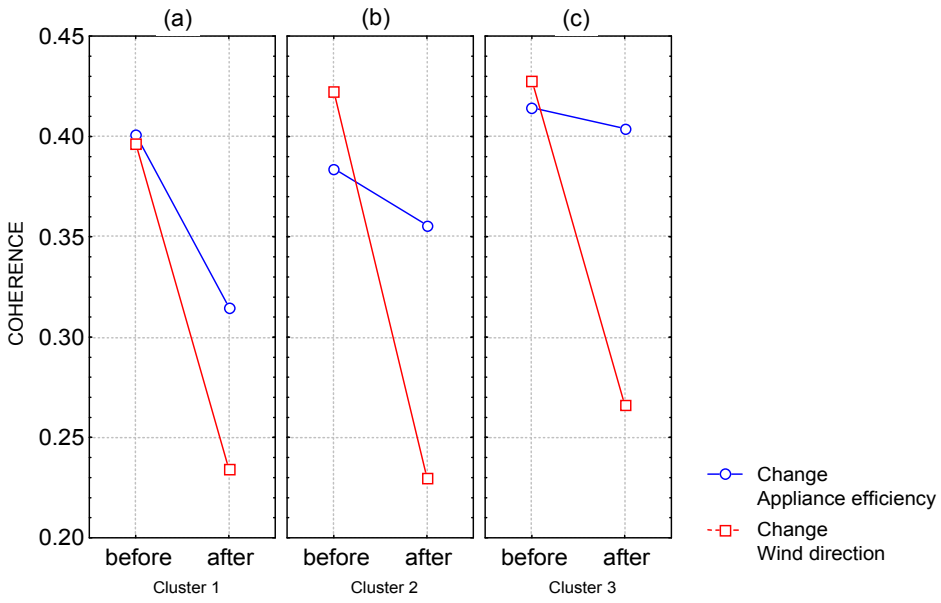
Figure 5 Interaction between environmental change type, before–after change, and cluster group using Coherence as dependent variable.


Trial 1
Trial 2
Trial 3
log files containing series of actions or states
type 1
type 2
…
type *n*
Types are states or actions


DROP VECTOR
CONTROL VECTOR
0.08
0.06
0.04
0.02
0
-0.02
-0.04
average cosine
1
2
3
cluster #


(a)
(b)
COHERENCE
OVERALL PERFORMANCE
0.45
0.40
0.35
0.30
0.25
0.20
0.85
0.80
0.75
0.70
0.65
0.60
before
after
before
after
Change
Appliance efficiency
Change
Wind direction


(a)
(b)
(c)
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
OVERALL PERFORMANCE
before
after
Cluster 1
Cluster 2
Cluster 3
Change
Appliance efficiency
Change
Wind direction


(a)
(b)
(c)
0.45
0.40
0.35
0.30
0.25
0.20
COHERENCE
before
after
Cluster 1
Cluster 2
Cluster 3
Change
Appliance efficiency
Change
Wind direction