A Matter of Design: A Proposal to Encourage the Evolution of Design in Instructional Design

Steven Schatz
Indiana University & Canteen Consulting

ABSTRACT

Kerr wrote of the black box of design decisions—that time when the "heart of an instructional solution first appears." Petroski writes of judgment as indispensable for engineering design decisions. Both instructional design and engineering are design sciences, yet engineering has a rich tradition of discussing and reflecting upon design decisions. Engineering shares a language by which they can discuss the decisions that underlay designs and, so may learn from failures and allow their practice to evolve. This paper suggests means by which instructional design may begin to develop a language of design.

Design is depicted as process involving three spheres. Sphere One—skills, Sphere Two—practice, and Sphere Three—application.

By developing a language to discuss and reflect upon Sphere Two processes, practitioners may share their practice, learn from each other, and encourage the field to evolve so that we may meet the changing needs of a knowledge economy. Herein, a proposal to encourage the development and use of a language to discuss Sphere Two decisions is detailed and initial steps are explored.

Businesses have fundamentally changed over the past two decades. With these changes come new challenges and opportunities for those involved in performance and training. In 1988, Peter Drucker wrote that in 20 years "...the typical business will be knowledge-based, an organization composed largely of specialists who direct and discipline their own performance through organized feedback from colleagues, customers, and headquarters." Thomas Stewart (1997) concurs that this change is occurring. He writes, "Knowledge is instilled in physical work, making it 'smarter'; knowledge work substitutes for physical; and more and more people are what might be classed pure knowledge workers." A "knowledge creating" company must continually innovate in order to cope with constant market evolution (Nonaka, 1991). In such an environment, when there is more information needed more quickly, when that information is in near-constant change, when timelines for developing instruction seem ever shorter, and when there is a greater need for information customized for particular users delivered just-in-time, it becomes ever more important to consider performance support interventions as adjuncts or replacements for training. However, while the instructional design field has established models that support the development and evaluation of training, we need models, which can help practitioners explore a range of interventions, which can meet the needs of the evolving workforce. In this paper, we turn to an examination of what Kerr calls the black box of design decisions (Kerr, 1983) and investigations of the design process in the fields of engineering and architecture to provide a direction toward the development models and the means to discuss and improve those models and our practice.

Kerr focuses on a crucial time in design—when a designer chooses which instructional approach to take. These decisions shape the intervention, setting an initial direction and continually directing the development. Kerr believes that the models of the field "make the procedure look deceptively easy…While algorithms for instructional design and decision tables attempt to provide a clear picture of just how the designer operates during those few crucial minutes or seconds when the heart of an instructional solution first appears." It is by looking within the processes that we as designers of instruction go through while making crucial design decisions that we may develop the potential for expansion and evolution of our field's practice. If we can investigate, discuss, and reflect upon these design decisions and evaluate decisions in light of results, we have a powerful tool for guiding and improving practice. Kerr looked for guidance to the design fields of art, artificial intelligence, and architecture. In this paper, we look to architecture and engineering.

Instructional design is often associated with engineering and other design sciences. Bern (1967), for example, argued that the engineering metaphor was increasingly literal and pertinent to education in the post-Sputnik era. He traced the roots of the idea of "educational engineering" to the 1920s. More recently, the works of Gibbons and Clark have reiterated the view of instructional design as a design science like engineering (Clark, 2002; Gibbons, 2000). Clark writes, "ISD is modeled after similar systems methodologies used in all professions that are based on design sciences, professions such as engineering and information technology."

Henry Petroski, an engineering professor at Duke, has written extensively on the processes by which engineers develop designs and analyze failure (Petroski, 1992a, 1992b, 1994). Petroski characterizes each engineering design as a hypothesis. When engineers design a bridge, they are constructing a hypothesis that a particular combination of materials will work in a specific situation, based on the lessons from past projects. What makes a design successful? Petroski emphatically states it is judgment, a term that mirrors Kerr's design decision.

> The first and most indispensable design tool is judgment. It is engineering and design judgment that not only gets projects started in the right direction but also keeps a critical eye on their progress and execution. .. It is judgment that separates the significant from the insignificant details, and it is judgment that catches analysis going astray…The single most important source of judgment lies in learning from one's mistakes and those of others. (Petroski, 1994 p.121)

When a bridge fails, engineers revisit the design decisions that were made to see what decisions (judgment) proved to be incorrect. Engineering has a language for discussing and evaluating what choices were made and what decisions led to the failure.

During the 1830s, it was common knowledge that suspension bridges could not be built so that they were stable enough to be seriously considered for heavy traffic. Suspension bridges had been built, and when they failed, engineers would discuss how they broke and why. These discussions included weight of the roadway, vibrations from wind or traffic (marching soldiers), different types of braces, and stiffness of material. John Roebling, who designed the Niagara Bridge and the Brooklyn Bridge, studied the failures and successes. After building the successful Niagara Bridge, he wrote "…what means have been used in the Niagara Bridge, to make it answer for Railway traffic? The means employed are *Weight, Girders, Trusses, and Stays.* With these any degree of stiffness can be insured, to resist either the action of trains, or the violence of storms, or even hurricanes" (ibid). Worldwide, engineers evidenced an interest in communicating the reason for failure and their design decisions by which they believed they could obfuscate failure. Such language permits reflective evaluation, and evolution of the practice of design.

It is by looking within the processes that we as designers of instruction go through while making crucial design decisions that we may develop the potential for expansion and evolution of our field's practice.

However, there are differences between engineering and instructional design. Success and failure are more apparent in engineering design than in instructional design. Even great designers will build very few bridges in their career. A project may take decades to develop, approve, and build. So, there is a great impetus to investigate and consider design decisions before, during, and after a project. Conversely, instructional designers have no clear measure of a failed intervention. Failure is never so obvious as a leaning or fallen tower, nor is it so clear-cut—rarely will an intervention not improve anyone's performance. There is also an economic disinclination to dig for failure—in a world where there is often only funding for level one evaluation, who (beyond academics) can afford to examine the results of an instructional intervention and then investigate the design decisions made so that one may reflect upon the impact of those decisions on the resultant success or failure of the intervention. In short, instructional designers will not, as one engineer did for a failed bridge, alter their travel plans to study the spectacular failure of an instructional intervention.

While an excellent designer, if pressed, could probably answer questions about why they chose to chunk the instruction as they did, why they included certain elements in the instruction and others in the performance guide, why the classes were scheduled as a three hour introduction and an online performance system or a five day boot camp or three weekends spaced a month apart, this kind of consideration is not part of the practice of instructional design. A recent ERIC search for instructional design and design decision resulted in 23 hits. Only Kerr's article actually addressed the processes of design decisions. This consideration is part of the black box into which Kerr was trying to look and key to opening the box is the development of a common language by which we can discuss our decisions, just as engineers (and, as we will soon see, some architects) do. By developing a language, which enables us to discuss the design of interventions, be they performance enhancements, or instruction, we may create a tool, which enables us to more deeply evaluate, reflect upon, and learn from our practice.

How can we go about developing such a design language? Both Kerr and Petroski make a distinction between skills, design decisions, and application to a specific task. Let us look at design as involving three spheres of activity (see Figure 1). Each sphere involves knowledge, processes, and decisions. Sphere One concerns skill and "tool" knowledge—whether is it a physical tool or a tool such as a task analysis or programming knowledge. This knowledge is independent of any particular project. Sphere Two decisions are also independent of a specific project. These are the generic processes, which guide design decisions—what types of interventions work in different circumstances. Sphere Three process and decisions involve the application of Sphere One skills and knowledge and Sphere Two design decisions to a specific problem. It may help to think of Sphere One as Skills; Sphere Two as Practice; and Sphere Three as Application. A more detailed look at each Sphere follows.

Sphere One processes and knowledge are the usual curriculum taught to aspiring practitioners. We can think of Sphere One as being what you need to know to "do" the work of a field. In engineering, this includes knowledge of materials, building processes, and understanding of the affects of environment on a structure. In building, this would include knowledge of tools. In instructional design, this would include learning theory, techniques for needs and task analysis, knowledge of media, methods to assess performance gap, and knowledge of different types of interventions. One may talk of Sphere One decisions (do I use a table saw or a skill saw) and Sphere One skills or knowledge (ability to use either type of saw). Most Sphere One knowledge is explicit. It involves both individual and group knowledge (Cook & Brown, 1999).

Sphere Three decisions and knowledge involve the process of applying skills and design decisions to a specific intervention. This is the area we can currently point to and reflect upon in our practice of designing interventions. Sphere Three knowledge may reside in all of Cook and Brown's four types of knowledge (individual/tact, individual/explicit, group/tacit, group/explicit) as well as the generative interaction between knowledge as a tool and knowing, the action of using that knowledge in a practice.

"Each of the forms of knowledge is brought into play by knowing when knowledge is used as a tool in interaction with the world. Knowledge, meanwhile, gives shape and discipline to knowing" (Cook & Brown, p. 393). Sphere Three processes produce the most obvious product of an intervention. Whether a training book, an instructor's guide, a CD ROM, a performance support sheet, a redesign of workflow, a management change, or providing better lighting to improve performance, Sphere Three involves creating the end product of the practice of performance improvement. This is what we point to when someone asks us what we do. This is the deliverable.

Sphere Two decisions are the design decisions inside Kerr's black box, a designer's Practice. When Roebling wrote of his four means (weights, girders, trusses, stays) by which he insured the necessary stiffness of a bridge, he was expressing a design decision not connected to a specific application. Similarly, such underlying decisions as: "competition makes them learn faster," or "15% training to 65% performance system," or "immerse them in a boot camp style," or

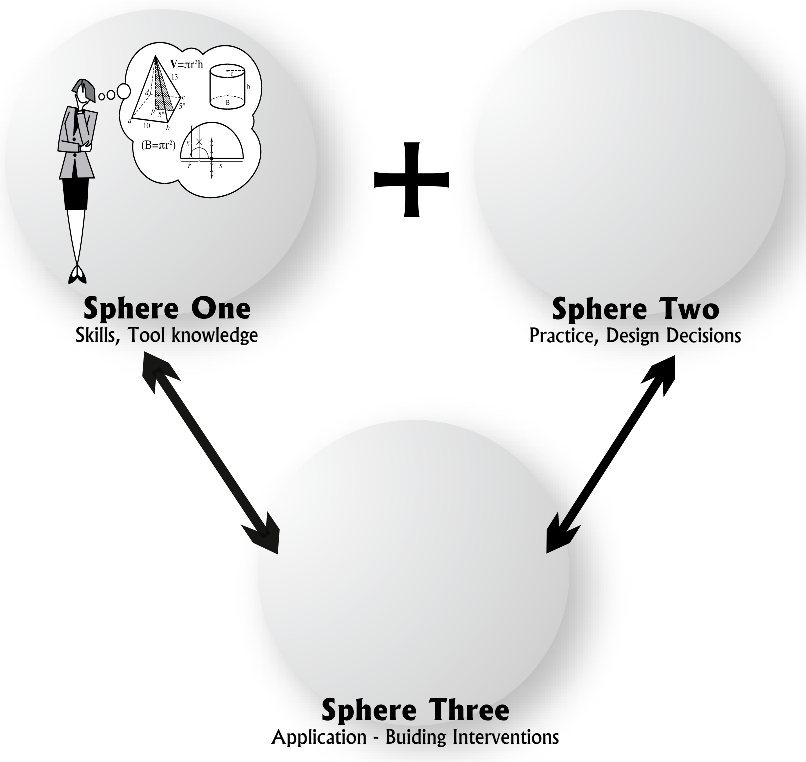

Figure 1. The three spheres of design.

“make it sexy and they will learn” are all design decisions not connected to a specific application, but are generic, Sphere Two decisions. Sphere Two involves stepping back from the specific application and making design decisions that shape the intervention. Where do we look for performance problems? What is most significant? What interventions will make the most difference? What blend of training, support, management, and environment interventions will provide the most effect for the best cost? What kinds of interventions will work with this population? All of these types of questions are Sphere Two decisions.

This is the region where skill and art mesh. These decisions are the reasons why an experienced and gifted practitioner will develop a more effective intervention than a neophyte will. Most of the Sphere Two knowledge is currently tacit/individual knowledge. It is our contention that if we could discuss and consider Sphere Two decisions, developing group/explicit knowledge of these decisions, the growth of knowledge in the field would be enhanced. Our field’s Practice would grow.

By viewing design in this manner and reflecting upon the decisions made in each of the three spheres of design, we have a new and powerful tool for reflecting upon and discussing interventions in order to learn from experience and obfuscate failure. For example, if you built a roof and there was a problem, Sphere One problems (might include bad shingles, not knowing how to cut properly, not knowing how to install shingles.) Sphere Three problems would include not following the plan and building too flat a roof, not overlapping the shingles enough (installers had the *knowledge*—Sphere One—but didn’t do it in this case—Sphere Three). Sphere Two problems would include planning a flat roof in a heavy rain area, not providing a place for the water to drain away. Sphere Two generic processes might tell you to consider drainage when planning a roof—flows and pitch. In instructional design, the division is not as conceptually simple, but may yield results that are more valuable.

By viewing design in this manner and reflecting upon the decisions made in each of the three spheres of design, we have a new and powerful tool for reflecting upon and discussing interventions in order to learn from experience and obfuscate failure.

If a performance intervention did not work, Sphere One problems would include poor task analysis, bad interviewing to find performance gaps, mistakes in graphics and text, programming problems, flaws in user testing methods. Sphere Three involves actually designing and building the instruction, so Sphere Three mistakes involve errors in how an intervention was planned and how it was "built". Was the writing, the graphic design, the implementation faulty? Did the instruction close the gap?

When evaluating interventions for Sphere Two problems, it is useful to distinguish between two types of processes—foundation and design. An example of a foundation type of Sphere Two decision is performance gaps identification (gaps identified as training gaps are often problems in communication, structure, or motivation). Sphere Two foundation type decisions include where to look, how to probe, and what to "smell for." While each project will result in different implementations, there are methods which all of us use, whether explicitly or not, to guide our investigation. Decisions of how much user testing to do is another Sphere Two foundation decision. The standard guide is "test to redundancy." When your data start repeating, you have discovered most of what you can get. In addition, formulaic approaches to instructional design, such as direct instruction (Engelmann, 1977) provide an excellent example of Sphere Two processes—generic methods that can be applied in multiple interventions.

Design types of Sphere Two processes offer guidance in the final product of an intervention. Examples include:

1. Develop to the audience—a technical audience usually wants specifics and details and suspect glossy presentation is hiding a lack of substance, a sales audience wants it fast and flashy.

2. Answer the underlying need. Often problems identified as training interventions actually are motivation or information problems. Look to the terminal result desired and develop an intervention that may not include any training.

3. Importance of face-to-face, informal interactions. A critique of online instruction is the lack of water cooler interactions. Thinking of the importance of face-to-face, one might design a course with a required onsite first meeting, or online informal communications exercises.

Such Sphere Two design type decisions and processes cannot be expressed as a single word. This is where we must begin to develop a language for discussing design. For guidance, we turn to the work of Christopher Alexander and his "pattern language." Alexander has done extensive work on the development of such design patterns for architecture. (Alexander, 1979; Alexander et al., 1977) Each pattern is several pages long. Each one is a window into a single recurring design decision, a direction, a consideration. In the form developed by Alexander, each pattern contains a context, a problem, a discussion, and a solution. For example, one pattern is "window place". In rooms that work, there is a window place, where one can stop and sit and look out of the window. There is a place to sit. There is something to see. Good rooms have a window place. It is a design decision to put such a place in a room. Perhaps the greatest value of Alexander's patterns is that, once written, they may be discussed and debated. Designers can discuss why they put a window in a particular place and the purpose it will serve. Alexander will be discussed in greater detail later. However, imagine the power for performance practitioners of being able to describe decisions made about the design of an intervention in such a way. To be able to evaluate, reflect upon, and learn from not only the end product, but also the Sphere Three foundation and design decisions that shaped and directed the design of the performance intervention would allow us to learn from successes and failures, to open the black box, to engage in what Cook and Brown call the generative dance.

Instructional designers usually learn Sphere One knowledge in school and do Sphere Three work in their career. They spend little time considering Sphere Two decisions. However, it is Sphere Two where most interventions will succeed or fail and it is by studying Sphere Two decisions that we have the greatest chance to learn and evolve as practitioners and as a field. We typically evaluate trainings by looking at Sphere Three end products. We currently only have a language for evaluating Sphere Three results. The problem with only looking at Sphere Three results is that it does not give a clear means to obfuscate failure. However, a Sphere Two evaluation of the foundation and design type decisions that led to the final product does provide a means for such reflection. For example, for the evaluation of a recent performance intervention, we used the following Sphere Two explanations:

> Stated goals were to create a knowledge base. We decided to conduct onsite observation and interviews. Questions focused on what killer app would drive them to use an online tool. Using Weick's sensemaking, evaluated interviews and observations. Conclusions—need was communications tools. Specifically: need for fast discovery of specialists, up to date inventory data, tools to support unofficial communications.

In instructional design, we do not have a common language for considering what Sphere Two decisions were made to construct an intervention, so even if we want to begin this type of conversation, we lack the means. We need a method...a language, which will enable us to discuss, to evaluate, and to learn from the Sphere Two design decisions. These decisions get to the heart of what Cook and Brown call genre (Cook & Brown, 1999) or group tacit knowledge. It is my belief that Sphere Two decisions lay at the heart of design and that it is crucial to the evolution of our field that we develop methods of discussing and learning from these Sphere Two decisions. A journey of a thousand miles begins with a single step. The following proposal offers a direction for our first steps in the development of the means and inclination to discuss Sphere Two design decisions.

A Proposal

Instructional design as a field needs to develop means to reflect upon, discuss, and communicate design decisions—to open the black box. The following activities would serve to encourage the development of the field as a design science, the ability to discuss the application of these processes and the recognition that reflection and communication is crucial to the continued growth of our field.

1. A field wide effort to develop a "language" for communicating the process of a project and analyzing successes and failures. This development should be a major goal

throughout the field and the practice of analysis of training products using this "language" should become standard practice. The work of Alexander and those in the software design community can guide this activity.

2. Continued emphasis of the importance of this effort and analysis to our field by:

a. Inclusion of such analyses as regular sections in major journals of the field.

b. Establishment of working groups on process language by ISPI (International Society for Performance Improvement), AECT (Association for Educational Communications and Technology), and AERA (American Educational Research Association) with reports issued throughout the year and public forums presented at conferences. Outreach/promotion by these groups at industry conferences including Online Learning and Training.

c. Development of a knowledge repository by a professional association (ISPI, ASTD, or AECT) of proposed and accepted patterns along with discussion areas to encourage consideration of patterns as they evolve.

3. Mandatory courses within major degree programs in instructional design that (1) teach the methods of analyzing practice and (2) use the methods to review and evaluate past and current instructional interventions for successes, failures and possible improvements.

4. Encouragement of "cross-cultural" development teams working on authentic development projects with students from other design fields including architecture, information design, engineering, and graphics with the expressed goal to be the development of group understanding of the design process—discussions of how decisions are made.

> ***It is my belief that Sphere Two decisions lay at the heart of design and that it is crucial to the evolution of our field that we develop methods of discussing and learning from these Sphere Two decisions.***

A Process Language

In order to discuss the design decisions we make while developing interventions, we must develop a common language to describe what we do during Sphere Two decisions. What did we identify as the main inhibitors of performance? Why did we decide on a training intervention? What other types of interventions did we consider? Why did we decide to use pictures to support one part of instruction and text only for another? Why did we decide to use humor to present a section? How many people did we interview to ascertain the population characteristics? How many assessment items were included at what points?

If we have the means to talk about these decisions, we can revisit these decisions to analyze what works and does not work in interventions. For guidance in the development of such a process language, we turn to the work of Christopher Alexander.

Mr. Alexander, an architect and faculty member at UC Berkeley has had far-reaching influence on engineering, building, social policy, computer programming and architecture. His ideas, put forth in two books, *A Timeless Way of Building* and *A Pattern Language* detail his group's attempts at developing such a language for the design of buildings. Alexander (1979), in his studies of buildings throughout the world has developed the following ideas:

1. There is a design that is right, correct for a given time, and place. It is possible to specify that correct design.
2. It is possible to develop a language of patterns that define a problem and a solution. These patterns, once developed can be discussed, argued about, amended and used. Their use can be reviewed and discussed.
3. We must design for ourselves. When we try to design for others, we stray from the useful to the fashionable.
4. Building is never complete; there is a continuous process of repair as needs of the users change.

Alexander provides a model for developing a language to talk about design. Can we begin to examine instruction or performance support with this eye? Is it possible to identify instruction or support tools that are "good"? Can we begin to identify what makes them good? What pieces must go together for our designs to work? Alexander demonstrates the use of patterns in designing a house and a collection of buildings. He demonstrates how using such a language allows a designer to talk about decisions made and to continually improve. This work provides a blueprint for the task that lies ahead and a promise of what is possible once the task of developing a language begins to generate results.

> Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice. (Alexander, 1977, p. x)

Let us look at a pattern from *A Pattern Language*, called "workspace enclosure" to get an idea of what a pattern looks like. This pattern is the part of the consideration Alexander thinks should go into the design of an office. This piece is the design of a particular workspace. The problem this pattern addresses is, "People cannot work effectively if their workspace is too enclosed or too exposed. A good workspace strikes the balance." A discussion of approaches, methods of discovery and conclusions too lengthy to reproduce here follows. Finally, a solution is offered in a way that can be applied in many different settings. It is followed by a rough sketch of a possible implementation as well as other patterns that are integrally connected to this pattern. The "solution" calls for an area of at least 60 square feet with significant window and openings. The person behind the desk should look out into a larger space with at least eight feet clear.

There should be a connection with other works, but never more than eight workspaces within view or earshot of one another (Alexander, 1977, p. 851).

We need to begin to identify patterns in the design of performance interventions. Alexander's patterns narrow in scope from regions, to cities, to neighborhoods, to structures, to building techniques. We might find patterns in each of the areas of systemic design—patterns which guide who to talk to when understanding the problem, patterns guiding what to look for, patterns guiding what and how much to ask, patterns to guide where to look for a solution. There will be hundreds of patterns which weave together to form performance interventions. For example, one pattern I use regularly is one called, "Killer App." The problem may be stated: People are busy. They do not want to use new systems. The consideration is: The best performance system in the world will do no good if it is not used. However, unless coerced, many new systems will not be used, even if they are useful. Sometimes we can coerce use. If it is a training intervention, we can take them away from their work and put them in a class. If it is a system, we may tie use to pay or other carrot/stick options. Sometimes we can try to tie the system in with other systems that are already used—adding a function to a portal, for example. People resist new things. They cost time, energy and may not work. There is a natural conservatism—if it ain't broke…don't fix it. The solution is: Look for the Killer App. The killer app is a term from computer evolution. It refers to the application that is so important, so useful, so compelling that it drives people to adopt a new system…a new way of doing things. Spreadsheets drove businesses to accept the need for personal computers. Email drove the development of the internet. Desktop publishing drove the adoption of laser printers. Browsers drove the adoption of the web. When considering a system, when talking to potential users, be searching for a killer app. What function could be included in the system that would save so much time, that would be so useful that it would drive users to take the time to learn to use the system, to open up yet another window on their computer. Always design at least one killer app in every intervention so that users will use what you build.

Patterns are more statements of policy than linear instructions. They may be applied differently in every instance, but provide a coherent approach to design.

Patterns are more statements of policy than linear instructions. They may be applied differently in every instance, but provide a coherent approach to design. The above example is given not to provide a straight jacket that must be copied, but to provide a starting point for the discussion of how to develop our own language of patterns. We must first begin to identify when we make design decisions, what problems we are answering at those points, and then see if there are recurring answers we use that result in good interventions.

Alexander's work has inspired software architects who attempt to build a pattern language for designing large software systems. Perhaps the most accessible book is Gabriel's Patterns of Software (Gabriel, 1996), but there is an extensive, technical literature detailing the efforts to develop pattern language for software developers. These examples can provide guidance as our field attempts to develop our own pattern language. Perhaps the most important lesson learned from software developers is that the process of developing a language has great value on its own. As practitioners discuss, argue and come to understandings of what "works" and what are the underlying principles, they develop a greater understanding of their practice. Weick calls this sensemaking (Weick, 1995), where meaning is developed through the process of reaching unified understandings. This holds great promise to our field. By undertaking the consideration of performance interventions, examining what design decisions have been made, and judging whether or not the intervention has proven useful, we begin to develop the language and the abilities to distinguish what design decisions are made. We begin to develop both means and inclination to examine our work as designers. If we do that, we will certainly begin to expand our methods for supporting performance, not based on what new technology is hot, but on effective design decisions. It hardly matters what form a language for talking about Sphere Two processes takes, what matters is that we, as a field begin to talk about it, to work toward it. The arguments that such discussions is certain to generate will build our knowledge and understanding of our practice…our genre. Secondly, the proposition offers actions that serve to elevate the importance of this work. We are academics and must make this effort important in areas we view as important—journals, conferences, classes and activities. We, as a field, must begin to explore and must share our explorations with each other. One of the main professional organizations should establish a knowledge base where a continuing dialog may take place—posting suggested and established patterns, discussion of uses and utility of such patterns, and suggested changes and new patterns. This may become a vibrant area of consideration and discussion. The goal is not the "Answer," but to consider the Question as a regular part of practice. This is the generative dance Cook and Brown talk of where knowledge (individual, group, tacit and explicit) and knowing interact to create knowledge. These considerations will certainly change over time (ongoing repair in Alexander's words), but will continue to help our field evolve.

Evidence of Need

There is ample empirical evidence of the lack of such considerations within our field and a need to develop such a language. Offered here are four very brief indications - a review of current literature, a review of some graduate programs in IST, experiences at conferences, and finally, a brief analysis of a recent large misadventure in the field all demonstrate the value of this exercise.

Review of Literature. Most of the literature in IST looks at *use* of artifacts instead of process of design. Most scholarly writings, even those specifically detailing an instructional tool, talk about the effect on learning or some other aspect of use. Nine journals published in 2002 were reviewed (Educational Technology Research & Development—2 issues, Performance Improvement Quarterly, Journal of Learning Science, Journal of Educational Computing Research, Journal of Educational Technology and Society, Journal of Research on Technology in Education, Interactive Learning Environments and Educational Media International). Seventy-one articles were examined for exposition of design considerations and methods. Of those 71, 62 offered no such information. Of the nine that did, two were new conceptualizations of the ISD model and only three offered real insight into design decisions. None talked about how to talk about such decisions.

Review of Programs. Similarly, a review of course descriptions for three Ph.D. programs in instructional technologies (Florida State University, Indiana University, Utah State University) showed no classes that might be construed to address Sphere Two considerations. There was marked similarity in programs. All offered theory, research classes, a class or two on systems and several on building interventions. There were no classes that discussed design theory across disciplines.

Industry Conferences. Industry conferences (Online Learning, Training, ISPI and ASTD) evidence great interest in design decisions. Attendees want to know specifics—problems, solutions, and design decisions. Theory is not as useful to practitioners as examples from the trenches. Some of the best-attended sessions during these conferences are "State of the Industry" presentations, which demonstrate a small group of cutting edge projects, complete with discussions of initial tasks, design decisions, development challenges, costs, and timelines. People in industry want and need this information. As a field, we must not turn away from this expressed need.

The Trouble with Learning Objects. Over the past several years, there has been a great interest in the concept of reusable learning objects. Millions of dollars have been invested. However, some very fundamental problems with learning objects have arisen. Beyond concerns about objectifying knowledge, it is becoming clear that the idea of objects brought together on the fly is pedagogically unsound. Learning objects cannot create good instruction because they offer information without context. There are potential benefits to the interest and work with learning objects. These include (1) encouraging the development of smaller, more modular bits of instruction; (2) allowing teachers or designers to bring together web-based elements more easily—a digital library of elements to use when building instruction; (3) objects being used to support performance—short pieces of information providing specific answers to user inquiries—instead of as instructional objects.

While these potential gains are noteworthy, they are a far cry from the initial utopian view of learning objects that foretold of learners being able to retrieve disparate bits on the fly based on individual assessment and preferences, netting huge savings and increases in quality. Why did so many believe the initial pipe dream? Why was it not obvious that, technical questions aside, learning objects could not create good instruction?

It was because we, as a field, are not clear about the processes designers engage in during Sphere Two. In the fantasy of learning objects, user testing and survey can give us Sphere One information—we can know the terminal result expected, the learner's knowledge/skill level, and user preferences. Sphere Three is taken care of—the pieces will be displayed on screen. However, we do not know enough about these Sphere Two decisions - the process of making the many design decisions between all the different options for a specific person in a specific setting to develop an effective intervention which may not even be a training intervention, but performance support, motivation or some other action. We cannot explain how these decisions are reached, so it would not be possible to tell a machine how to make them. In addition, the more we consider what those Sphere Two processes involve, the more clear it becomes that they cannot be done on the fly. As Alexander writes:

> In short, no pattern is an isolated entity. Each pattern can exist in the world, only to the extent that is supported by other patterns: the larger patterns in which it is embedded, the patterns of the same size that surround it, and the smaller patterns which are embedded in it.
>
> This is a fundamental view of the world. It says that when you build a thing you cannot merely build that thing in isolation, but must also repair the world around it, and within it, so that the larger world at that one place becomes more coherent, and more whole; and the thing which you make takes its place in the web of nature, as you make it. (Alexander et al., 1977, p.xiii)

Learning objects cannot work. Had we known more about the practice of designing performance interventions, this would have been apparent several years ago.

Conclusions

Instructional design/performance improvement is a design field that does not discuss the work of design, the black box. It functions now as an information gathering field with the Sphere Two decisions being made individually, with no consistency and with little sharing of process, knowledge, or learning. Currently, there are no means to communicate, discuss, and reflect upon the process of design. Because of this, there is no way to go back and learn from failure, so there is no impetus to discover failure. The result is a field that is still locked in the techniques developed decades ago. Instead of improving design, we have become promoters of new toys to present the same techniques with more flash and sizzle.

In order to evolve, we must learn to talk about design. We must take action to develop methods to reflect upon Sphere Two processes, to truly evaluate the design of our interventions. It must become standard practice to explain interventions by the design decisions made and to evaluate designs in the same way. If we take on this challenge, there will be some lovely rows as we finally begin to open the black box. As we become more accustomed to discussing design, we may discover that we can have greater impact developing interventions that explore performance improvements other than training. Whatever the result, we can explain for the first time what we are doing, why we have made the decisions we have made and discuss, analyze, and learn from successes and failures. The field may evolve, meeting the changing needs of business. No longer in danger of becoming marginalized in a knowledge economy where jobs often change too quickly to offer an adequate return on development costs, we may be able to offer real solutions to current and future problems. These are actions worth taking.

References

Alexander, C. (1979). *The timeless way of building*. New York: Oxford University Press.

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A pattern language*. New York: Oxford University Press.

Bern, H.A. (1967). Wanted: Educational engineers. *Phi Delta Kappan, 48*(5), 230-236.

Clark, R.C. (2002). The New ISD: Applying cognitive strategies to instructional design. *Performance Improvement, 41*(7), 8-15.

Cook, S.D.N., & Brown, J.S. (1999). Bridging epistemologies: The generative dance between organizational knowledge and organizational knowing. *Organization Science, 10*(4), 381-400.

Drucker, P.F. (1988). The coming of the new organization. *Harvard Business Review, 66*(1), 9.

Engelmann, S. (1977). Direct instruction. In C.R. Dills & A.J. Romiszowski (Eds.), *Instructional Design Paradigms* (pp. 371-390). Englewood Cliffs, NJ: Educational Technology Publications.

Gabriel, R. (1996). *Patterns of software: Tales from the software community.* Oxford: Oxford University Press.

Gibbons, A.S. (2000). *The Practice of instructional technology.* Paper presented at the Annual International Conference of the Association for Educational Communications and Technology, Denver, CO.

Kerr, S.T. (1983). Inside the black box: Making design decisions for instruction. *British Journal of Educational Technology, 14*(1), 45-58.

Nonaka, I. (1991). The knowledge creating company. *Harvard Business Review, 69*(6), 96.

Petroski, H. (1992a). *The Evolution of useful things.* New York: A. Knopf.

Petroski, H. (1992b). *To engineer is human: The role of failure in successful design.* New York: Vintage Books.

Petroski, H. (1994). *Design paradigms: Case histories of error and judgment in engineering.* Cambridge: Cambridge University Press.

Steward, T.A. (1997). *Intellectual capital: The new wealth of organizations*. New York: Doubleday/Currency.

Weick, K.E. (1995). *Sensemaking in organizations*. Thousand Oaks: Sage.

STEVEN SCHATZ specializes in solutions design at the intersection of people, performance, and technology—effective use of meta-tagged performance objects, online communication/collaboration, performance support systems, and KM strategies. Completing a Ph.D., his focus is search/retrieval of information in performance systems. *E-mail:* schatz@powerstart.com