8-2010

A Design Science Based Evaluation Framework for Patterns

Stacie Clarke Petter
University of Nebraska at Omaha, stacie_petter@baylor.edu

Deepak Khazanchi
University of Nebraska at Omaha, khazanchi@unomaha.edu

John D. Murphy
University of Nebraska at Omaha

A DESIGN SCIENCE BASED EVALUATION FRAMEWORK FOR PATTERNS

Abstract

Patterns were originally developed in the field of architecture as a mechanism for communicating good solutions to recurring classes of problems. Since then, many researchers and practitioners have created patterns to describe effective solutions to problems associated with disparate areas such as virtual project management, human-computer interaction, software development and engineering, and design science research. We believe that the development of patterns is a design science activity in which an artifact (i.e., a pattern) is created to communicate about and improve upon the current state-of-practice. Design science research has two critical components, creation and evaluation of an artifact. While many patterns have been created, few, if any, have been evaluated in this sense. In this paper, we propose a framework to evaluate patterns in any domain and provide examples of how to use the evaluation framework. This process of evaluation could help researchers refine extant patterns and improve the possibility of creating sustainable best practices for a given domain. We believe this evaluation framework begins an important dialogue related to the evaluation of patterns as artifacts of design science research. We draw upon the literature associated with patterns, evaluation of design science research, and research methods to develop this framework for evaluating patterns in a more consistent and rigorous manner.

Keywords: Patterns, pattern languages, design science, evaluation.

A DESIGN SCIENCE BASED EVALUATION FRAMEWORK FOR PATTERNS

INTRODUCTION

Patterns are a useful method to describe a good solution to a recurring problem (Alexander, 1999; Alexander et al., 1977). Originally created for architecture by Alexander (1964), the use of patterns has become increasingly popular for many domains, including those within the computing field ~~within information systems (IS)~~. Examples include software engineering (Gamma et al., 1995), virtual project management (Khazanchi & Zigurs, 2006), human-computer interaction (Tidwell, 2005), and design science research (Vaishnavi & Kuechler, 2007). Developing patterns in a given domain offers something interesting and unique to practitioners and researchers. For practitioners, patterns offer practical and applied knowledge by providing high-level solutions to classes of problems that can be converted into specific best practices. For researchers, patterns can provide a method to synthesize and capture knowledge in a given domain as well as highlight areas for future research.

Within design science research, researchers develop artifacts to solve a problem and evaluate their utility (Hevner et al., 2004; March & Smith, 1995; Vaishnavi & Kuechler, 2008). Patterns can be considered an artifact of design science research in which the artifact that is created (i.e., the pattern) is helping a practitioner solve a problem. Patterns have been developed for a wide variety of domains based on the "best practices" derived from experts, yet they are rarely evaluated after being documented. New and existing patterns that have not been rigorously evaluated can present many questions. Does the documented pattern veridically represent the knowledge gained from experts? Is the pattern applicable beyond the immediate context of the knowledge provided by the expert(s)? Was the pattern documented in a manner that is understandable beyond the pattern creator?

By conducting a formal evaluation of the pattern artifact, consistent with design science research, researchers have the opportunity to refine and enhance new and existing patterns. This evaluation is based on the premise that patterns exist at various states in their evolution in a "pattern life-cycle". As patterns evolve from being formalizations of mere speculation, heuristics and experiences, to well articulated, substantiated descriptions of practices, they become more valid and useful. Patterns in the early phases of their life cycle are "raw," and those at later phases are more refined and valid. Thus, the nature of our evaluation approach and rigor may also vary depending on the state of the pattern. Evaluation of patterns at various phases in their life-cycle has the possibility to provide additional insight about the pattern, underlying theory, and the domain. These insights can be instrumental to practitioners hoping to take advantage of the pattern in that there could be more information provided about the possible implementation of the pattern, boundary conditions for using the pattern, as well as empirically founded results that could be expected from the use of the pattern

Therefore the purpose of this paper is to propose and demonstrate a formal approach to evaluating patterns with appropriate consideration of a pattern's state in its life-cycle. This need is highlighted by the results of a recent study wherein a pattern set in the human-computer interaction domain was examined for its utility (Wania & Atwood, 2009). Wania and Atwood found little evidence to support the touted benefits of patterns in a laboratory setting. More support for the use of patterns in human-computer interaction appeared in a more naturalistic context. It could be that the utility of individual patterns may not be completely realized until they have been refined through rigorous evaluation.

Within the current literature on patterns, we have many examples of patterns that have been created, but we have little evidence that their utility has been evaluated in a consistent and rigorous manner with consideration of the pattern's life-cycle. How does one vouch for the usability, reliability and validity of a pattern? We propose that patterns can be evaluated using criteria derived from the long-established scientific method and the literature related to research methodology, specifically the evaluation of constructs, variables, and measures (Cook & Campbell, 1979; Khazanchi, 1996; Straub et al., 2004), complemented with more recent evaluation techniques that are emerging from design science research (Baskerville et al., 2007; Vaishnavi & Kuechler, 2008).

The organization of the rest of this paper is as follows. First, we briefly describe the design science approach to research, focusing on key design science evaluation methods, and discuss some of the history and literature associated with patterns. We then synthesize key ideas from those two domains to construct a general framework that guides the development and explanation of our evaluation criteria. From that baseline, we present specific guidelines that we derive for evaluating patterns and apply those criteria to two existing patterns from two different domains. We offer this framework for evaluating patterns not as a comprehensive approach, but as the starting point for a discussion for better methods to more rigorously evaluating this particular artifact. Finally, we discuss implications of this research for practitioners and researchers and offer concluding thoughts on future research.

BACKGROUND

Design Science

Design science research in information systems "creates and evaluates IT artifacts intended to solve identified organizational problems" (Hevner et al., 2004, p. 78). Design science can be described as an approach that "aims at developing ways to achieve human goals" (March & Smith, 1995, p. 254). The goal of design science research is to create artifacts that offer value and utility (Hevner et al., 2004) as opposed to explanation or behavioral research, which attempts to discover and validate theory to explain the world (March & Smith, 1995).

The first activity in design science research is building the artifact to solve a problem in practice. There are many types of artifacts that can be created in design science research such as design theories, constructs, models, methods, and instantiations (Gregor & Jones, 2007; March & Smith, 1995; Vaishnavi & Kuechler, 2008; Walls et al., 1992). In developing and building an artifact, the researcher draws upon theoretical knowledge, contextual information, organizational needs, and personal creativity (Hevner et al., 2004; March & Smith, 1995; Vaishnavi & Kuechler, 2008). Yet, simply building an artifact to address a problem is not enough to constitute a research activity (Nunamaker et al., 1990-1991).

The second design science research activity is evaluating artifacts. Evaluation research can be described as "an attempt to assess the worth or value of some innovation, intervention, service, or approach" (Robson, 2002). There are many different approaches to evaluation, just as there are many different types of artifacts, yet the purpose of evaluation remains consistent. It should also be noted that though evaluation cannot be achieved without a purpose, it is not an end in itself. The goal of evaluation is to answer questions such as, "Does the artifact or theory work?" and "How useful is the artifact or theory?" The benefits of evaluation are threefold. First, evaluation of the artifact is necessary to confirm that the artifact is feasible and offers some improvement over current practice (Nunamaker et al., 1990-1991; Vaishnavi & Kuechler, 2008). Second, the evaluation activity offers feedback to the researcher to identify if the problem is well understood, if the assumptions are appropriate, if the quality of the design process is appropriate, and if there are needed refinements for the artifact (Hevner et al., 2004). Thirdly, once an artifact has been evaluated one can use social science research approaches to theorize or explain why the artifact did or did not work in a particular environment (March & Smith, 1995).

In design science, artifacts are evaluated by identifying a set of assessment criteria for the environment in which the artifact is to be evaluated and examining how well the artifact meets the specified criteria (March & Smith, 1995; Nunamaker et al., 1990-1991). When specifying the assessment criteria and evaluation process, the researcher should not only consider the type of artifact, but also the context and environment in which the artifact will be used (March & Smith, 1995). There are many different methods to evaluate an artifact and can range from descriptive techniques that use logic and scenarios to demonstrate the utility of the artifact to empirical methods such as experimentation and case studies (Hevner et al., 2004). Evaluation can occur either based on the design specifications (i.e., *ex ante* evaluation) or after its implementation (i.e., *ex post* evaluation) (Pries-Heje et al., 2008). Evaluation techniques can be artificial (i.e., contrived settings such as simulations or laboratory experiments) or naturalistic (i.e., realistic settings such as case studies and action research) (Venable, 2006a). Some have argued that it is important to evaluate artifacts using both hard evaluation methods, similar to those used in positivist research and theory testing, as well as soft evaluation methods, based on soft systems methodology which considers the context and human component that can affect the evaluation of the artifact (Baskerville et al., 2007). In this paper, we support a holistic approach to evaluation of the artifact of patterns in which pattern evaluation can occur both *ex ante* or *ex post* (Pries-Heje et al., 2008), artificially or naturalistically (Venable, 2006a), and using either hard or soft evaluation methods (Baskerville et al., 2007).

PATTERNS

> *Each pattern describes a problem which 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 et al., 1977)*

The concept of patterns was originally developed for architecture by Alexander (1977). In general, ~~the~~ pattern theory provides an intuitive way to comprehend complexity and communicate about it. According to Khazanchi and Zigurs (2006), patterns are analogous to ideas such as genre taxonomies (Yoshioka et al., 2001), recurring themes, familiar processes, rules of thumb, or standard procedures. Alexander (1964) argues that patterns should describe ”good” solutions to problems so that all essential qualities are captured and can then be used in other contexts.

As stated in the opening quote, Alexander defines a pattern as a three-part rule that expresses a relationship among a specific context, a problem, and a solution. The problem is a set of forces that occur repeatedly in that context. The solution is a certain “spatial configuration” that allows the set of forces to resolve themselves. The pattern itself describes how the solution can be used whenever the problem occurs in that particular context. A collection of patterns represents a pattern language, defined as a system of patterns that combines to produce a variety of important outcomes (Alexander et al., 1977).

Alexander's notions of patterns is used in many domains within the computing discipline, such as programming (Beck & Cunningham, 1987; Gamma et al., 1995), human computer interaction (HCI) (Tidwell, 2005), and computer supported cooperative work (Fernandez et al., 2002; Schuemmer, 2003). In addition, these notions have appeared in the domain of management (Austin & Westerman, 2002), group leadership (Homsky, 2003), project management, and virtual project management (Khazanchi & Zigurs, 2005). In the project management domain, Völter (2002) introduces the concept of "anti-patterns" which document all the wrong responses to project management problems. Moreover, patterns have also been developed for conducting design science research to aid researchers in understanding approaches for finding relevant problems to address, stimulating creativity, and publishing research (Vaishnavi & Kuechler, 2007). In our subsequent discussion, we will use an example from the field of virtual project management and an example from human-computer interaction to illustrate our evaluation approach. It is clear that we, as researchers, have been quite good at creating artifacts in the form of patterns and pattern languages for problems and contexts in various domains. However, what we could improve upon is the evaluation of this particular type of artifact.

DEVELOPING CRITERIA TO EVALUATE PATTERNS

CURRENT PRINCIPLES FOR CREATING PATTERNS

To date, there is some agreement on the principles for creating patterns. First and foremost, patterns have a certain format, which includes a meaningful name, a problem statement, the context for the problem, the applicable forces and constraints, a solution, one or more examples, the context after the pattern has been applied (which may include side effects), the rationale, a listing of related patterns, and known uses of the pattern (Appleton, 2000).

Alexander (1979) has also argued that patterns should have a "quality without a name" (i.e., QWAN) in which there is an aesthetic beauty, durability, order, and emotional resonance. He describes this concept as:

> *This oneness, or the lack of it, is the fundamental quality for anything. Whether it is in a poem, or a man, or a building full of people, or in a forest, or a city, everything that matters stems from it. It embodies everything.*
>
> *Yet still this quality cannot be named.* (Alexander, 1979, p. 28)

While this quality is difficult to describe or name, there are facets inherent to QWAN that can potentially be evaluated that tell us whether a pattern is really good. We believe that even though it may be difficult to evaluate patterns, we can provide guidelines and ideas for researchers to better evaluate patterns as part of the design science research paradigm.

There is an understanding within the pattern community that feedback is necessary to improve and validate patterns (Brown et al., 1998). Feedback can be provided by those that have implemented the pattern or by experts that have read the pattern (Brown et al., 1998). One set of researchers identified three primary criteria for evaluating patterns within the human-computer interaction domain: 1) are the users "empowered" through the use of patterns; 2) does the pattern allow users to "generate complete designs"; and 3) does the pattern produce "life-enhancing outcomes" for the user (Deardon et al., 2002). Patterns have also been described as "good" if they offer a proven solution to a problem that considers system structures and the human component (Coplien, 2007).

To date, the evaluation principles that are available are useful mostly for creating and developing patterns (e.g., Lea, 1994; Meszaros & Doble, 1996). However, this level of evaluation is not as robust as it could be, and furthermore it appears that many patterns are not evaluated in any kind of systematic manner. Currently researchers may propose one or more patterns and offer limited evaluation. Our approach attempts to formalize principles from the design science and construct/conceptual validation literature to develop a more systematic means of evaluating patterns.

TOWARD NEW EVALUATION CRITERIA OF PATTERNS

From a design science perspective, researchers have proposed many different approaches to evaluating artifacts. For example, traditional research techniques such as case studies, field studies, and experimentation can be used (Hevner et al., 2004; Robson, 2002; Vaishnavi & Kuechler, 2008). Design science researchers evaluating artifacts also have the option of other evaluative techniques that use logic, description, or simulations (Hevner et al., 2004; Vaishnavi & Kuechler, 2008).

Researchers conducting traditional behavioral research, rather than design science research, often have an evaluative component in their research design. For example, positivists evaluate measurement instruments and constructs in their quest to identify and predict underlying truth (Trochim, 2000). Interpretivists evaluate their interpretation and understanding of a phenomenon (Klein & Myers, 1999). Considering the importance of evaluation in many paradigms used in information systems and the varying forms of evaluation, we suggest that pattern evaluation should blend "hard and soft" techniques (using the terminology of Baskerville et al., 2007). "Hard" techniques of pattern evaluation are more objective in their assessment and interpretation, while "soft" techniques are subjective and consider the effects of human factors and the context under study. Thus, we seek to develop criteria using a holistic approach that considers all potential forms of evidence to validate patterns.

Toward this end, our first step in developing pattern evaluation criteria was to draw heavily from extant research in the area of research methodology in the social sciences. Evaluation often uses empirical techniques, such as those used in the natural and social sciences (Venable, 2006a). Examples of these works included Straub (1989; 2004), Cook and Campbell (1979), Sproull (1995), and Hevner et al. (2004). Most of this work was oriented around improving the construction and application of measurement items to better establish the validity of constructs (Cook & Campbell, 1979; Straub, 1989; Straub et al., 2004). However, other research has examined the idea of validation at different levels of abstraction that ranged from proving the truth of theories (Iivari, 2007) to validating concepts (Khazanchi, 1996). Distilling these readings, evaluation in its purest form could be understood as the process of assuring that any "component" (i.e., theory, concept, construct, survey item or pattern) contains all of the features inherent to its individual definition and that it fulfilled its intended purpose.

Using this background, we identify key criteria from across design and explanatory research that could be useful in evaluating patterns. Individual criterion that appeared in multiple sources were given special attention as those ideas were deemed to have stronger support, but the goal was to adapt important validation concepts into analogous evaluation criteria framed within the context of design science and patterns. The notion of validity carried through the "classic" IS research refers to whether or not a construct or model represents the underlying truth (Cook & Campbell, 1979; Robson, 2002); however, this is not the purpose of adapting these different forms of validation for evaluating patterns. Patterns, while they are artifacts like theories or other constructs, are still different from other research constructs. Patterns exist as a means of deriving useful solutions to recurring problems within specific contexts. As such, one pattern can only be more or less useful than another, not more or less true, so our ultimate goal is to assess the viability of the pattern. However, this does not take away from the fact that patterns and their documentation have the potential to veridically represent the properties ascribed to the best practices, experiences, etc. that is incorporated in the "good" solution to a problem within a given context. Thus the need for systematic evaluation of patterns becomes even more critical.

Evaluation Throughout the Pattern Life-Cycle

In proposing criteria for evaluation of patterns we argue for evaluation throughout the life-cycle of a pattern primarily because (i) pattern discovery (Khazanchi & Zigurs, 2007), pattern description and validation are concomitant activities -- the better our pattern descriptions, the better the pattern language, greater the acceptability of the patterns, and in turn, greater the possibility that the patterns are valid; and (ii) aA pattern that is plausible, feasible, effective, etc. is more likely to be imputed with the same meaning and properties to different users and thus potentially result in better utility from a design science perspective.

It is our view that evaluation of patterns needs to be a conscious continuous improvement activity as is true of design science approaches in general. Our criteria of evaluation make sense of this by looking at pattern evaluation in the context of its entire life-cycle. Like all human endeavors, writing of patterns in any form is susceptible to human fallibility and biases. How do we verify (ensure with some degree of certainty) that the pattern described by an author captures the "essential good qualities" imputed to a solution described by a pattern (Alexander, 1979)? Mere representation of an expert opinion in the form an Alexanderian pattern is no better than conjecture until it is actually used and yes, evaluated. Evaluating patterns in a life-cycle attempts to get to this by making them better (or more useful).

Khazanchi and Zigurs (2007) citing Alexander's (1979) notion of "good qualities" embodied in patterns argue that if we can capture these essential qualities of what we do well (and what we do not do well), then we can apply these qualities to do the same thing in other contexts and applications. Further citing Alexander they assert that these "good qualities" can only be captured by observation, by experience, by both positive and negative examples and, at times, by abstract argumentation (Alexander, 1979). In this way, we can "… discover some property which is common to all the ones which feel good, and missing from all the ones which don't feel good" (p. 255). Candidate patterns can be discovered via induction from experience or knowledge or by mere speculation about what works or does not or through deduction (Khazanchi & Zigurs, 2007). If we accept this argument, "candidate" patterns can only have plausibility to start with and need to be further evaluated to establish incremental validity (Khazanchi & Zigurs, 2007). Our pattern evaluation life-cycle attempts to set the context in which a candidate pattern can be improved beyond mere plausibility.

Another important consideration when evaluating patterns arises from the fact that there is a natural life-cycle to patterns. As described earlier, patterns emerge from lessons learned in the practice of a particular discipline. Domain experts accumulate these lessons and season them with knowledge earned through study of the domain's theoretical base. These experts are then able to shape and reshape patterns which can be re-used in the domain. Together, these activities constitute the development of a pattern. But to be useful, the pattern has to be adapted to the specific context of its intended use. This deployment phase consists of carefully analyzing the situation to identify boundary conditions and constraints and then developing a specific instantiation of the general solution described in the pattern. With careful crafting, a pattern can then be put to use where specific results can be compared against intended effects described in the pattern definition. This cycle of development, deployment, and use continues, as depicted in Figure 1.

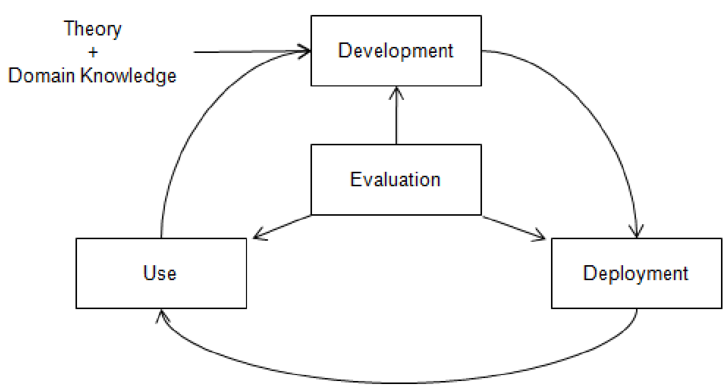

Figure 1: Pattern Life-cycle

Patterns can be evaluated at each step in this life-cycle and so should continue to improve, just as theories continue to improve through the scientific research cycle (as theory improvement is described in Straub, 1989). Certain risks are likely in each phase of the life-cycle. There are some inherent risks that can occur in the various phases of development for design science artifacts, and these risks are also likely to also occur in the life-cycle of a pattern. Just as a design science research project could be jeopardized by risks such as a poor description of the problem, lack of understanding of the problem, or failure to apply appropriate knowledge to the problem (Baskerville et al., 2008), these risks could also occur during the development phase of the pattern life-cycle. In design science research, real risks occur in creating a hypothetical solution that cannot be implemented, does not work as intended, or one that causes additional problems (Baskerville et al., 2008). These risks are consistent with those that occur in the deployment segment of the pattern life-cycle. Implemented solutions that do not work as intended is a risk for both design science research (Baskerville et al., 2008) as well as in the use phase of the pattern life-cycle.

These three phases of the pattern life-cycle offer unique opportunities for evaluation, and in fact it may prove in practice that certain phases are more amenable to certain forms of evaluation than other phases. For example, in the development phase, evaluation approaches such as expert review and literature review would be most appropriate because the pattern has not been operationalized or used as of yet. For patterns that are being operationalized, but have not been applied yet, instantiating the pattern in a generic context or a prototype would be appropriate. In addition, peer review would also be a possibility for patterns in this phase of the pattern life-cycle. Finally, patterns that are being used can be subjected to observational methods (qualitative and quantitative) to examine the utility of the pattern in a specific context.

Proposed Pattern Evaluation Criteria

Table 1 is a list of the evaluation criteria that we derived from the literature, the traditional definition of the concept in the validation and evaluation literature, and its applied definition within the context of patterns.

Table 1: Criteria for Evaluating Patterns

Evaluation Criteria	Traditional Definition	Adapted Definition for Patterns
Plausible	The degree to which a concept is more than just a belief (Khazanchi, 1996; Sproull, 1995).	Pattern is sensible considering the current understanding of the domain (Alexander, 1979; Brown et al., 1998; Khazanchi & Zigurs, 2007).
Effective	The degree to which a concept describes the phenomenon under study parsimoniously and stimulates inquiry (Khazanchi, 1996; Rossiter, 2002; Straub et al., 2004).	Pattern is described in language that is understandable; root causes of the problem are identified and addressed by the recommended solutions (Appleton, 2000).
Feasible	The degree to which a concept is workable or operationalizable (Khazanchi, 1996).	Pattern can be operationalized or implemented as described.
Predictive	The degree to which a concept is capable of predicting outcomes for given conditions (Khazanchi, 1996; Sproull, 1995; Straub et al., 2004; Trochim, 2000).	Pattern produces the expected result or produces a result in the intended direction (Coplien, 2007).
Reliable	The degree to which a concept is certifiable by different researchers using different methods (Campbell & Fiske, 1959; Khazanchi, 1996; Straub, 1989; Straub et al., 2004).	Pattern produces similar results regardless of the implementer or technique (Baskerville et al., 2007).

Exemplars

To demonstrate how these criteria can be used in evaluating patterns, we will use one of the patterns from Khazanchi and Zigurs' *Patterns of Effective Management of Virtual Projects* (2005). Their book postulated that the management of virtual projects is fundamentally different from traditional project management and synthesized best practices to develop patterns appropriate for managing virtual projects. Virtual project management patterns exist in a domain that has a high level of complexity and abstraction.

Many extraneous factors unrelated to the pattern being implemented can potentially affect the management of virtual projects. The specific pattern that we will use to illustrate the application of our criteria is "Managing Virtuality." We selected this particular pattern because it is an example of a pattern that would be difficult to formulaically evaluate due to the number of possible confounds and the difficulty of isolating the effects of the pattern. Khazanchi and Zigurs (2005) define this pattern as follows:

Pattern for Managing Virtuality (Khazanchi & Zigurs, 2005)
Context Your team is having difficulty with time zone differences at both the national and global levels. This is particularly highlighted during crunch time or crisis situations when communication is not prompt, delaying the problem resolution process. **Problem** How do you overcome time zone and geographic differences and effectively engage all team members? **Solution** Overcome distance barriers due to time zone and geography by eliminating then during activities requiring intensive interaction and coordination, such as project initiation, by temporarily collocating team members. Require periodic site visits and travel by team members to different sites. Designate team member liaisons as focal points of coordination who spend some time in the home office location, to become acculturated and informed about technical issues; liaisons can then transfer knowledge to local sites for day-to-day coordination. Assign team members in one geographic region (e.g., North and South America) to tasks requiring telephone or video-based interactions because they share time zones and thus can more easily schedule conferences. **Discussion** Collocating team members for face-to-face interactions can also help then establish ground rules and common understanding that facilitate communication and coordination when team members return home. This also allows team members to build a social network and stimulates the development of team identity, cohesion, and commitment that can potentially be sustained once team members are again dispersed (Davidson & Tay, 2003). Overcoming time zone differences is critical not just for global teams. For example, in the USA, one of our study participants stated that project notifications from the pacific time zone would reach the central time zone later in the day, leaving less time or required work outside of normal hours for team members.

The HCI domain has also embraced the use of patterns and several different pattern sets exist. HCI patterns have been developed to format content, guide users through actions, and present an aesthetic interface (Tidwell, 2005). We use one of these patterns, “Liquid Layout”, to illustrate the evaluation framework because in contrast to the virtual project management pattern, it is purportedly more concrete or tangible. The implementer has more control on the execution and evaluation of this type of pattern as compared to the virtual project management pattern.

Liquid Layout (Tidwell, 2005)[1]
What:[2] As the user resizes the window, resize the page contents along with it so the page is constantly "filled." **Use when:** The user might want more space -- or less -- in which to show the content of a window, dialog box, or page. This is likely to happen whenever a page contains a lot of text (as in a web page), or a high-information control like a table or tree, or a graphic editor. It doesn't work as well when the visual design requires a certain amount of screen real estate, and no more or less. **Why:** Unless you're designing a "closed" UI like a kiosk, a handheld, or a full-screen video game, you can't predict the conditions under which the user views your UI. Screen size, font preferences, other windows on the screen, or the importance of any particular page to the user -- none of this is under your control. How, then, can you decide the one optimal page size for all users? Giving the user a little control over the layout of the page makes your UI more flexible under changing conditions. It may also make the user feel less antagonistic towards the UI, since he can bend it to fit his immediate needs and contexts. If you need more convincing, consider what happens to a fixed-layout, "non-liquid" UI when the language or font size changes. Do columns still line up? Do pages suddenly become too wide, or even clipped at the margins? If not, great; you have a simple and robust design. But pages engineered to work nicely with window size changes generally also accommodate language or font size changes. **How:** Make the page contents continuously "fill" the window as it changes size. Multiline text should wrap at the right margin, until it becomes ten to twelve words wide (more on that later). Trees, tables, editors, etc. at Center Stage should enlarge generously while their margins stay compact. If the page has anything form-like on it, horizontal stretching should cause text fields to elongate -- users will

[1] This pattern can also be viewed at http://designinginterfaces.com/Liquid_Layout. This website also contains pictures of the concept discussed in this pattern.
[2] Although the headings in the Liquid Layout pattern differ from the headings of the Managing Virtuality pattern, both patterns exemplified contain all of the components required for a pattern as defined by Appleton (2000) and Alexander et al. (1977).

Liquid Layout (Tidwell, 2005)[1]
appreciate this if they need to type in anything longer than the text field's normal length. Likewise, anything scrolled (lists, tables, etc.) should lengthen, and usually widen too. Web pages and similar UIs should allow the body content to fill the new space, while keeping navigational devices and signposts anchored to the top and left margins. Background colors and patterns should always fill the new space, even if the content itself cannot. What happens when the window gets too small for its content? You could put scrollbars around it. Otherwise, white space should shrink as necessary; outright clipping may occur when the window gets really tiny, but the most important content hangs in there to the end. If you deal with paragraphs of text, remember that they become nearly illegible when they're too wide. Graphic designers target an optimal line length for easy reading of text; one metric is 10 to 12 average English words per line. Another metric is 30 to 35 em-widths -- that is, the width of your font's lowercase "m". When your text gets much wider than that, users' eyes have to travel too far from the end of one line to the beginning of the next one; if it gets much narrower, it's too choppy to read easily. (That said, there is evidence that text with a longer line length, such as 100 characters per line, is faster to read than shorter lines, even though users prefer to read lines fewer than 55 characters long. See the guidelines at http://usability.gov, especially "Use Reading Performance or User Preference.") A well-behaved Liquid Layout can be difficult to implement in complex web pages, especially if you want to dispense with tables and use straight CSS. It's also hard in Visual Basic and Visual C++ (or was, at least, for a long time). However, Java provides several layout managers you can use to implement it. **Examples:** Mac OS X allows you to resize the standard "Open" dialog box, which uses a liquid layout. This is good because users can see as much of the filesystem hierarchy as they want, rather than being constrained to a tiny predetermined space. (The Cascading Lists pattern is used in this dialog, too.)

We specifically chose exemplar patterns from virtual project management and human computer interaction to illustrate the fact that patterns may be relatively more abstract or more concrete. Our goal was to identify patterns from two completely different pattern sets that address very different types of problems. Using these two patterns, we want to illustrate that our evaluation framework is appropriate across pattern domains. The pattern from virtual project management represents patterns that have been developed for domains such as management or conducting design science research. These types of patterns illustrate a situation in which "soft" evaluation methods are more likely needed to account for the abstract nature of the pattern. Since the second pattern from the HCI literature is more concrete and consistent with design patterns in software engineering, "harder" evaluation methods may be more readily available than for those patterns that are more abstract. It is important to note that we understand that both hard and soft techniques can be used for evaluating patterns, regardless of their level of abstraction. We have used extreme examples of patterns to illustrate the robustness of our pattern evaluation framework.

CRITERIA FOR EVALUATING PATTERNS

PLAUSIBLE

The first attribute that a pattern should possess is plausibility, which traditionally examines whether a concept is reasonable, is not just mere belief or conjecture, and could be demonstrated by existing research/practice within a domain (Khazanchi, 1996). The idea of plausibility is analogous to face validity wherein one explores whether the pattern is coherent and sensible within the framework of established domain knowledge (Sproull, 1995). Since patterns are descriptions of good solutions to classes of problems that are captured by observation, experience, and through positive and negative examples (Alexander, 1979; Khazanchi & Zigurs, 2007), they should possess a degree of believability that goes beyond mere supposition or conjecture (Khazanchi, 1996).

By definition, patterns should be derived from practice; however, some candidate patterns may not have adequate plausibility because they are mere speculation based on personal insight or experience. Within the context of patterns, a plausible pattern is one that makes sense given current understanding of the domain. In some sense, a pattern is conferred with plausibility by the fact that patterns are higher-order abstractions of good solutions to problems embedded in real experiences of domain experts. To determine if a pattern is plausible, one should ask three primary questions. First, is the pattern *prima facie* reasonable for its proposed use? This face validity is a minimum condition for plausibility. In general, it may be presumed that patterns by their very nature have high face validity and thus some degree of plausibility since they describe solutions that are derived from an analysis of expert opinion about a problem class. Second, is the pattern believable beyond conjecture? In posing this second question, we expect that a pattern will have higher plausibility if there is evidence to indicate that it is representative of the practices and/or processes that constitute the domain. Part of this evaluation also includes consideration if the problem is indeed worthy of being documented. Brown et al. (1998) suggest applying a "rule of three," suggesting at least three experiences or occurrences of the problem are necessary to identify if the problem in interesting and if the solution works. Finally, one should ask if the pattern fits within the realm of the problem domain. At times a pattern may initially appear to be inconsistent with other established domain patterns, but this could simply be due to circumstances peculiar to the application of the pattern. Under such conditions, when there is a reasonable justification for the pattern to be logically inconsistent with other patterns in the domain, then this should be clearly explained within the pattern's description.

Pattern plausibility is best evaluated during its development phase. Thus, the first level of screening for plausibility comes through the process of building the pattern: in the judgment of its creators, is the pattern a reasonable representation of "good" solution for the problem being addressed. Typically, a pattern is derived from practice and is representative of a problem and solution that has been observed to work multiple times.

Patterns may be developed using anecdotal, rather than empirical, evidence. Although the form of data used to develop the pattern may not be rigorous, the review of the pattern by experts for its reasonableness within the domain is still an important aspect of establishing plausibility. Furthermore, the development of a pattern may include some level of examining literature to determine what is already known about the problem, and this literature review will help validate that others have identified aspects of the pattern too (Khazanchi, 1996; Straub et al., 2004). Additional approaches to evaluating the plausibility of a pattern could include the use of thought experiments (Weick, 1989) to intellectually inspect the pattern.

EXEMPLAR OF PLAUSIBLE CRITERION

As explained in the methodology portion of *Patterns of Effective Management of Virtual Projects* (Khazanchi & Zigurs, 2005), the Managing Virtuality pattern was derived from ideas generated in virtual focus groups of established project management professionals. Therefore, the first fundamental questions of plausibility have been addressed: the pattern is *prima facie* reasonable and the pattern is more than conjecture and contains all facets that are necessary to fully describe it. Furthermore, the pattern is supported by literature as noted by the citations included within the pattern itself. We have also examined the pattern along with other patterns associated with virtual project management. The problem and solution are consistent with the other patterns, also contributing to the plausibility of the pattern. Now that the pattern (along with others in this domain) has been published and disseminated through multiple channels, its plausibility can be furthered through examination by additional experts.

The Liquid Layout pattern also meets the criteria for plausibility. Inspection of the pattern shows that it is a reasonable practice within the HCI domain, particularly when creating web layouts, where users typically have the capability to resize the screen. Ensuring that the user is able to read the contents of the page regardless of the choice for screen size is, on the surface, reasonable. This pattern also answers the second question which asks if the pattern is believable beyond conjecture. First, as is described in the pattern description, it is possible to apply this pattern in many application environments. Furthermore, this pattern was developed nearly ten years ago and has been published on a well-known and well-cited website and in a book on the topic of HCI. Many experts in the domain have had the ability to view and comment on the pattern, especially if a better solution to the problem discussed in the pattern is available. Finally, the pattern is consistent with the principles of developing user interfaces, satisfying the third criteria of plausibility.

Effective

Effectiveness is a measure of the degree to which a concept is parsimoniously, yet completely, represented and stimulates further inquiry (Khazanchi, 1996). The idea of effectiveness encompasses the idea that the concept is described completely. In measurement theory, content validity seeks to assess if the measures for a construct are thorough and without errors of inclusion or exclusion (Straub et al., 2004). A clear definition of the concept or construct is required to assess if the representation is indeed effective (Rossiter, 2002).

In the context of evaluating patterns, we combine the above notions and describe effective patterns as those that are complete while being economically communicated. Appleton (2000) described a complete pattern as one that includes a meaningful name, a problem statement, the context for the problem, the applicable forces and constraints, a solution, one or more examples, the context after the pattern has been applied (which may include side effects), the rationale, a listing of related patterns, and known uses of the pattern. Including each of these components helps the user understand the pattern adequately enough to fully and properly implement the solution. In addition to being complete, effective patterns will be clear, internally consistent and correct.

Even though patterns are derived from practicing experts we cannot state with a degree of certainty that they are always effective. This is because the patterns are "described" in Alexandrian form by someone who may (a) not communicate the essence of the pattern well, (b) exclude things that are important in the formation of the pattern description, (3) ignore facets that are not clear or understandable, (4) not capture correctly the timeless quality (QWON) of patterns that makes them independent of changing technology or other factors, and (5) generalize inadequately "what is good" in the solution beyond the descriptions provided by a few experts.

A pattern's effectiveness can be assessed during the development and deployment phases of the pattern's life-cycle. During development, the pattern's author(s) can compare their description to a checklist of required components developed through literature review, such as that provided by Appleton (2000). The pattern's effectiveness could also be scrutinized by a separate panel of experts, using procedures similar to assessing content validity (Rossiter, 2002; Straub et al., 2004). Moving into the next phase of a pattern's life-cycle, observing users as they attempt to deploy a pattern is another excellent opportunity to assess a pattern's effectiveness. Their actions will indicate how easily and accurately they understand the information conveyed in the pattern description. Of course every situation will be different in some regards, and these differences may drive variations in how the pattern is deployed, but the spirit of an effective pattern should to some degree shine through in its intended use. Over time, pattern descriptions will change to accommodate newly discovered boundary conditions – in effect, becoming more effective. Sometimes new patterns will emerge when the changes are deemed too great to fit within the existing pattern description.

Exemplar of Effective Criterion

As described above, evaluating the effectiveness of the Managing Virtuality pattern starts with considering the clarity of the pattern's description. The Managing Virtuality pattern includes all components recommended by Appleton for pattern descriptions (2000). The clarity of the pattern's description is reflected by two key facts -- the pattern was developed by data collected from and reviewed by project management experts, and the pattern was included in a research book that was reviewed by editorial staff.

The Liquid Layout pattern underwent a different development phase that also allowed vigorous assessment by a large number of evaluators. The original pattern was previewed on the developer's personal website during the late 1990s where it was available for review for literally a world-wide panel of subject matter experts. The pattern was progressively updated and subsequently posted on another website devoted to user interface patterns and techniques[3] where it continued to be available for review and use by the world-wide community. The pattern was then published in the author's book, *Designing Interfaces: Patterns for Effective Interactive Design* (Tidwell, 2005). As such, the pattern has been distilled for nearly a decade by innumerable experts. This particular pattern could be further evaluated by asking students designing a web interface to read the pattern and incorporate the concepts from the pattern within the website design. An

---

[3] http://~~time-tripper.com/uipatterns/~~www.designinginterfaces.com

expert could evaluate the students' prototypes of the website and determine if the novices were able to understand the essence of the pattern.

Feasible

Feasible concepts have the characteristic of being workable (Khazanchi, 1996). This notion can be extended to the evaluation of patterns wherein feasible patterns are those that can have the quality of being implementable or operationalizable. Thus going beyond mere plausibility of a pattern, feasibility ensures that a pattern can be actually used. A key consideration in assessing feasibility lies in identifying and articulating boundary conditions that could possibly constrain implementation. Examples of boundary conditions that might render a pattern impractical are policies, regulations, and ethical or physical constraints that preclude successful implementation of the pattern's recommended solution(s). The presence of these circumstances does not necessarily render a pattern infeasible as one might be able to tailor the pattern's implementation to accommodate specific contexts. The question that this evaluation criterion attempts to get at is whether a user of the pattern can take a plausible pattern and specify an instantiation of it. Thus, a feasible pattern possesses a fundamental sense of viability, which includes a manageable set of boundary conditions. A feasible pattern should ensure successful implementations for the preponderance of the pattern's instantiations.

This criterion could be evaluated in several ways at different points in the pattern's life-cycle. First, patterns are developed as a means of capturing the knowledge of domain experts about how problems have been addressed successfully in the past. This development process inherently conveys some degree of feasibility to the pattern from its earliest instantiation that can go beyond just plausibility. As the pattern is being crafted, the developer(s) could also conduct a series of thought experiments to mentally examine the feasibility of the pattern under a variety of boundary conditions (Weick, 1989).

After the pattern has been developed, either the pattern creator or a separate panel of experts could then review the pattern's feasibility. Depending on the nature of the pattern and the situation it is designed to address, it may also be possible to mathematically or logically argue for a pattern's feasibility. Another potential evaluation option in the deployment phase of the pattern life-cycle is to conduct a small-scale prototype implementation of the pattern to observe whether the pattern can actually be implemented. Specific evaluation measures available in the deployment phase include interviews with those implementing the pattern or peer reviews by other experts to examine if the pattern will indeed be operationalizable within the specific context of its intended use.

Exemplar for Feasible Criterion

In examining the Managing Virtuality pattern in terms of this criterion, one can conclude that this pattern is derived from ideas garnered from a panel of domain experts who participated in a series of virtual focus group discussions. The pattern was then further assessed and streamlined by the team of project management researchers who wrote the book in which the pattern was included. Finally, the pattern is now published so that others can examine it and provide feedback on their ability to operationalize the pattern and identify new boundary conditions found under a broader range of implementation scenarios.

The Liquid Layout pattern has existed on various instantiations of the developer's user interface (UI) Patterns Techniques websites [4], where it has been available for review and use world-wide since the late 1990s. It has also been published in a book entitled *Designing Interfaces: Patterns for Effective Interaction Design* (Tidwell, 2005) that has been reviewed by numerous subject matter experts. In terms of deployment, the pattern has been used for nearly a decade by website designers in a wide-range of development environments, and that use on innumerable websites stands testimony to the feasibility of its recommended solutions. While its use has been widespread, the pattern does state that it is more difficult to implement for complex web pages, Visual Basic, and Visual C++. These are the boundaries of this pattern and the author of the pattern has appropriately identified them for the potential user.

PREDICTIVE

Within measurement theory, predictive validity strives to demonstrate that the measures for a specific construct is able to predict another construct (Sproull, 1995; Straub et al., 2004). At the concept level, a concept is predictive if a given set of antecedent conditions can be expected to trigger a corresponding phenomenon (Khazanchi, 1996). Related to this idea is internal validity, which is the degree to which one can rule out alternative explanations for the results (Trochim, 2000).

We consider a pattern as having the attribute of being predictive if the implementation of a pattern produces the expected result and the use of the pattern is the reason for the result. To evaluate if a pattern is predictive, the researcher should simply assess if the pattern produces the expected result or not (Coplien, 2007). That being said, a

---

[4] http://~~time-tripper.com/uipatterns/~~www.designinginterfaces.com

pattern may not yield exactly the same magnitude of results every time it is applied because the context of application will always be somewhat different, and those contextual factors may dampen or amplify the effect. But while the magnitude of the effect will vary with the circumstances, the pattern should produce the same general effect each time it is applied. For instance, if the pattern was designed to improve communication within a virtual team, it should improve communication every time despite the fact that the amount of improvement will vary depending on the team's circumstances. Moreover, patterns are not variance models where more of a pattern necessarily translates into more of a result. The second evaluation of the predictive attribute for a pattern is examining if the pattern is the cause of the result. If a pattern is used, but an extraneous factor contributed to reaching the anticipated result, we cannot necessarily suggest that a pattern is predictive. The use of the pattern may not have been necessary to reach the outcome and therefore, may lack utility.

To evaluate the predictive nature of a pattern, the pattern must be used. This is not an attribute that can be evaluated in any other phase of the pattern life-cycle. Ideally, a researcher could perform a longitudinal study and observe practitioners using the pattern within their organization to determine if the pattern is predictive. Another approach could be to survey practitioners that have used the pattern in the past and ask them if the pattern indeed produced the intended result and if the practitioner perceives that the pattern is the reason for the result. Ideally, evaluation of this criterion will best occur as patterns are used and results from such use are isomorphic with the description of the pattern.

When examining patterns in these naturalistic settings to determine if the artifact works (Venable, 2006b), it is also appropriate to consider the social and organizational contexts that may affect the application of the pattern (Baskerville et al., 2007). If a pattern does not yield the precise expected result, one or more social factors may be culprit rather than a faulty pattern. Understanding the context when applying and evaluating the pattern can lend further insight should an unanticipated result occur. Furthermore, the pattern domain may also have an effect on the pattern's predictive ability. Technical and objective patterns, such as patterns for programming objects, are likely to achieve predictive validity with little difficulty. However, patterns that are more subjective and human-oriented, such as those created for conducting research or managing projects, may be more difficult to evaluate success or failure of meeting the expected result. Failure to hit a precise target may not mean a failure of the pattern if the pattern brings us closer to the desired result. The fuzzier the domain and patterns, the more necessary it is to consider the social and organizational contexts when evaluating predictive validity (Baskerville et al., 2007).

Exemplar of Predictive Criterion

With the publication of the Patterns of Effective Management of Virtual Projects book (Khazanchi & Zigurs, 2005), the Managing Virtuality pattern moves into a testable form, which is the heart of the predictive validity criterion. Now, as this pattern is applied in various organizations, it will become possible to gather empirical results from its instantiation and use within specific contexts, which can be used to confirm that its solutions do in fact address the problem of managing time zone and geographic differences to effectively engage all team members.

The evaluation of the Liquid Layout pattern is more straightforward. To rigorously examine if the pattern is predictive, a developer (or series of developers across contexts and applications) can use the pattern when developing the user interface. If the intended result (i.e., the user has the ability to control the contents of the page) is achieved, the pattern is predictive. To make the evaluation more objective, a usability assessment could be performed to confirm that the intended result was achieved. However, this more rigorous examination is probably not necessary for this particular pattern because it already has been implemented several times and examples are shown within the pattern. This particular pattern is already known to be predictive in that using the pattern 1) produces the ability for the user to have more control over the interface and 2) is the cause for the user to be able to exert more control over the interface.

Reliable

According to measurement theory, measures that make up a construct are reliable when there is evidence that "the respondent can answer the same questions or close approximations the same way each time" (Straub, 1989). Khazanchi (1996) describes how the ultimate test of a concept is having empirical data to corroborate it, and this notion applies to patterns too. Consistent with the concept of reliability in measurement theory, concepts should also be intersubjectively and intermethodogically certifiable meaning that different investigators using different techniques can obtain similar results each time a concept is applied (Khazanchi, 1996). This is consistent with measurement theory, which suggests using multiple methods to examine a construct to reduce the possibility of common methods bias (Campbell & Fiske, 1959; Straub et al., 2004).

In the context of patterns, meeting the reliability criterion requires that one gather evidence that confirms (or disconfirms) that the pattern consistently produces a close approximation of the desired effect. Furthermore, regardless of the implementer or operationalization of the principles embedded in a pattern, the results should be consistent. To evaluate if a pattern is reliable, the evaluator could assess if the pattern produces similar results regardless of the user of the pattern. Furthermore, the results should be reasonably consistent regardless of whether the pattern is evaluated using qualitative or quantitative assessment techniques.

The reliability of a pattern cannot be determined in the development or deployment phase of the life-cycle. However, multiple individuals could implement the pattern in real settings and then examine the results through both quantitative and qualitative techniques. This is analogous to the *test-retest* approach for evaluating reliability of constructs (Straub et al., 2004). In this test-retest evaluation, the pattern is reused multiple times in similar scenarios to ensure that the same type results emerge from that pattern application. If different individuals are implementing the pattern, then the pattern is intersubjectively certifiable. If different evaluation methods are used (i.e., different qualitative and quantitative methods), then the pattern can be deemed intermethodologically certifiable.

In more human-oriented patterns, particularly patterns that address human behavior and preferences, it may be difficult to achieve consistency, so it is critical to consider the context of the pattern application when evaluating these types of patterns for reliability (Baskerville et al., 2007). In this situation, reliability can still be examined, but considered contextually and tied to achieving the essence of a pattern rather than an exact result. This conscious approach to understanding the context of the pattern can provide greater insight to researchers on the problem as well as some additional issues for practitioners to consider when applying the pattern.

Exemplar of Reliability Criterion

Evaluating if the Managing Virtuality pattern is reliable hinges on the publication of *Patterns of Effective Management of Virtual Projects* (Khazanchi & Zigurs, 2005) and broad use of the pattern. Since patterns are presumed to reflect good solutions relating to a problem in a specific context, the idea is to instantiate that solution in multiple contexts to ensure that it works consistently. To date, reliability of the pattern has not been assessed in a true empirical sense with a researcher asking multiple individuals to implement the pattern and then observe the results. That said, previous research has examined the use of several of the techniques recommended as possible solutions in this pattern and found positive results (Maznevski & Chudoba, 2000). Given that this pattern is human-oriented and there are many environmental factors that can impact the pattern's success, when assessing if the pattern is reliable, it is important to measure and consider contextual factors that may impact the application of the pattern. While quantitative techniques can be used, it may be useful to also consider gathering qualitative data to complement the findings. The qualitative data may shed light on important boundary conditions or other factors that affect the use and consistency of the pattern.

To assess the reliably of the Liquid Layout pattern, one could implement the pattern in a variety of contexts. For example, most websites, developed in HTML or other web development environments, provide pop-up windows that reflect this principle. Visual Basic and Java also include functionality to allow developers to use this pattern. Therefore, the results can be seen consistently in a variety of contexts. For a more robust evaluation of reliability, the pattern could be tested in a variety of applications, such as websites, operating systems, games, and utilitarian information systems for businesses. If different individuals can use the pattern in a variety of environments and achieve the result of allowing a page to be filled, then the pattern can be considered as reliable. Yet, given the extensive use of this principle and pattern in the development of computer interfaces, it seems reasonable that there is enough anecdotal evidence in a variety of contexts by many different organizations (as stated within the pattern itself), to declare the pattern as reliable.

SUMMARY

Table 2 provides a summary of the evaluation criteria proposed in this discussion along with examples of questions one should ask when applying the criteria as well as when during the pattern life-cycle it would be appropriate to evaluate the given criteria.

Table 2: Evaluation of Patterns Considering the Pattern's Life-Cycle

Evaluation Criteria	Definition for Patterns	Application of Criteria	Evaluation Life-Cycle (Potential Evaluation Techniques)
Plausible	Pattern is sensible considering the current understanding of the domain	▪ Is pattern believable beyond conjecture? ▪ Does the pattern fit in the general realm of the problem domain? ▪ Does the pattern seem reasonable for its proposed use? ▪ Is it derived from practice?	Development (expert review, literature, consistent with practice)
Effective	Pattern is described in language that is understandable; root causes of the problem are identified and addressed by the recommended solutions.	▪ Is the pattern stated in terms that are precise, comprehensible, complete, internally consistent, and concise?	Development (expert review, literature, consistent with practice) Deployment (instantiation via prototype, peer review)
Feasible	Pattern can be operationalized or implemented as described	▪ Can the pattern be implemented as described? ▪ Can the basic pattern be tailored for specific situations/contexts?	Deployment (instantiation via prototype, peer review)
Predictive	Pattern produces expected result	▪ Does the pattern lead to better results? ▪ Does the pattern produce the intended result? ▪ Did the pattern's recommended solutions elicit the result?	Use (observational methods)

Evaluation Criteria	Definition for Patterns	Application of Criteria	Evaluation Life-Cycle (Potential Evaluation Techniques)
Reliable	Pattern produces similar results regardless of the implementer or technique	▪ Can others use the pattern and get similar results? ▪ Are the results similar when examined with differing evaluative techniques?	Use (observational methods)

Using the Pattern Evaluation Framework

We posit that evaluation of a pattern can occur at each phase in the life-cycle of a pattern. By performing this evaluation, a researcher has the opportunity to refine and improve the pattern further. When converting expertise into a pattern, there is the risk of misunderstanding the problem or misrepresenting the solution, therefore, this process of evaluation enables reflection about the pattern across the life-cycle of the pattern. When any of the evaluation criteria are examined, the results should be incorporated into the pattern to either help refine or enhance the pattern further. These results could be stated in the narrative of the solution if using Alexandrian form of describing patterns. The Alexandrian description of patterns can be expanded to include a new section that specifically articulates the evaluation that has been performed to date and the results.[5]

This framework can be used by either those that create patterns or those that implement and use patterns in their research or practice. Both groups can contribute to the confirmation and/or enhancement of the pattern. In view of this, the evaluation criteria we have proposed do not necessarily have to be examined in the order presented. When a researcher is using this framework, it is important to consider where the pattern is in the pattern life-cycle. For example, if the pattern is still in development phase, then it

[5] We thank an anonymous reviewer for this suggestion.

would be inappropriate to evaluate for reliability until the pattern could be instantiated and deployed.

However, there are some key limitations of this framework for evaluating patterns. First, we have defined a number of questions that can be asked of patterns across domains; however, the approach used to evaluate a specific pattern may vary based on the context. More concrete patterns may benefit from harder, more positivistic evaluation techniques while more abstract patterns may require softer or subjective forms of evaluation. Within our framework, we hesitate to prescribe specific evaluation techniques for each of the evaluation criteria because patterns vary dramatically both within and across domains. In our attempt to provide more rigor in evaluating patterns, we do concede that there is subjectivity in the choice of evaluation techniques and interpretation of the results. However, we believe our framework provides a more consistent approach to evaluating patterns than our current approaches. Also, patterns are human descriptions subject to the same fallibility associated with other similar endeavors.

A second limitation of our framework is that we focus on evaluating a single pattern rather than a pattern set. A pattern by definition is connected to other patterns, so an update to one pattern can potentially affect related patterns. While some of the criteria in our framework could be extrapolated to examine the pattern set in its entirety, we believe that given the sparseness of evaluation of patterns, it is reasonable and manageable to first focus on the evaluation of single patterns before evaluating an entire pattern set. Once there is an understanding of the best approaches to evaluate patterns, then these principles could be adapted to examine the pattern set as a whole.

Finally, we acknowledge that the evaluation criteria in this framework may not be comprehensive. However, there is a need to start somewhere in the development of evaluation criteria. We hope that as the framework is applied to the evaluation of patterns, there will be an evaluation of this framework and refinement as it proceeds through its own life-cycle. We see this work as a beginning and not the final word on the evaluation of patterns.

CONCLUSIONS

In this paper we have described how patterns have been used in a variety of domains to develop solutions to recurring problems. Understanding that design science research encompasses both the design and evaluation of an artifact to address a specific problem faced by practitioners (Hevner et al., 2004; March & Smith, 1995), we suggest that within the pattern development literature, researchers often build patterns but fail to evaluate them. While some standards exist to define what can constitute a pattern, few criteria exist for evaluating the utility of patterns.

In this spirit, we have presented an initial set of criteria derived from classical scientific method and principles that we believe could be useful in evaluating patterns. The pattern may be evaluated by the researcher who developed the pattern or subsequent researchers or practitioners who have an interest in the pattern itself. Researchers interested in creating patterns can focus on plausibility and effectiveness as they document and develop patterns identified in practice. The pattern creator can go on to evaluate the pattern further for feasibility, predictability, and reliability or other researchers and practitioners can take on these additional forms of evaluation when the pattern is deployed and used. Having, and using, such criteria gives researchers a process to conduct a rigorous evaluation of the utility and validity of patterns. This type of assessment can increase the level of confidence in a pattern or pattern language. Furthermore, this type of evaluation may then improve the use of patterns in practice and help to avoid re-solving recurring problems that occur in practice.

In this work, we use principles for evaluation of artifacts, concepts, and constructs and apply these ideas to the context of patterns. We do not dispute the point made by Beck et al. (1996) and Prechelt et al. (2002) – both write about design patterns in software engineering – who suggest that patterns can be used to effectively communicate complex concepts in software engineering. However, validity as we describe it is a totally different issue. Maybe these patterns are already passed the stage of plausibility and feasibility but clearly a reader may not be sure of this. Our pattern life-cycle evaluation process is one way of certifying this. Our intent in this paper has been to start a much needed dialogue within the pattern community on how to actually evaluate patterns. Our next step will be to apply these criteria to existing patterns to see how they hold up in a field application. Future research may include converting these guidelines into patterns so that we could then "eat our own dog food" by evaluating our evaluation patterns using our approach. Throughout this work, our goal is to maintain a dialogue in the field to identify better and more inventive methods to evaluate the artifact of patterns within design science research.

References

Alexander, C. (1964). *Notes on the Synthesis of Form*. Cambridge: Harvard University Press.

Alexander, C. (1979). *The Timeless Way of Building*: Oxford University Press.

Alexander, C. (1999, Sept/Oct). The origins of pattern theory: the future of the theory, and the generation of a living world. *IEEE Software, 16,* 71-82.

Alexander, C., Ishikawa, S., and Silverstein, M. (1977). *A Pattern Language*. New York: Oxford University Press.

Appleton, B. (2000, Feb 14). Patterns and Software: Essential Concepts and Terminology. Retrieved Jan 9, 2007, from http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html

Austin, R. D. and Westerman, G. (2002). *Destiny WebSolutions, Inc.*Unpublished manuscript, Boston.

Baskerville, R., Pries-Heje, J., and Venable, J. (2007). "Soft design science research: Extending the boundaries of evaluation in design science research," *2nd International Conference on Design Science Research in Information Systems & Technology*, Pasadena, CA.

Baskerville, R., Pries-Heje, J., and Venable, J. (2008). "Evaluation Risks in Design Science Research: A Framework," *International Conference on Design Science Research in Information Systems and Technology*, Atlanta, GA.

Beck, K., Crocker, R., Meszaros, G., Vlissides, J., Coplien, J. O., Dominick, L., and Paulisch, F. (1996). "Industrial Experience with Design Patterns," *International Conference on Software Engineering*, Berlin, Germany, 103-114.

Beck, K. and Cunningham, W. (1987). "Using Pattern Languages for Object-Oriented Programs," *OOPSLA Workshop on the Specification and Design for Object-Oriented Programming*, Orlando, FL.

Brown, W., Malveau, R., McCormick, H., Mowbray, T., and Thomas, S. W. (1998, June 2, 1998). The Software Patterns Criteria: Proposed Definitions for Evaluating Software Pattern Quality. 10.2. Retrieved July 20, 2009

Campbell, D. T. and Fiske, D. W. (1959). "Convergent and Discriminant Validation by the Multitrait-Multimethod Matrix," *Psychological Bulletin,* Vol. 56, No. 2, 81-105.

Cook, T. D. and Campbell, D. T. (1979). *Quasi-experimentation: Design & analysis issues for field settings*. Chicago, IL: Rand McNally College Publishing Company.

Coplien, J. O. (2007). A Pattern Definition: Software Patterns. Retrieved Jan 8, 2008, from http://hillside.net/patterns/definition.html

Deardon, A., Finlay, J., Allgar, L., and McManus, B. (2002). "Evaluating Pattern Langugages in Participatory Design," *Conference on Human Factors in Computing Systems*, Minneapolis, MN, 664-665.

Fernandez, A., Holmer, T., Rubart, J., and Schuemmer, T. (2002). *Three Groupware Patterns from the Activity Awareness Family*. Paper presented at the Seventh European Conference on Pattern Languages of Programs.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented systems*. Reading, MA: Addison-Wesley.

Gregor, S. and Jones, D. (2007). "The Anatomy of a Design Theory," *Journal of the Association for Information Systems,* Vol. 8, No. 5, 312-335.

Hevner, A., March, S., Park, J., and Ram, S. (2004). "Design Science in Information Systems Research," *MIS Quarterly,* Vol. 28, No. 1, 75-105.

Homsky, O. (2003). *More Patterns for Group Leadership*. Paper presented at the Eigth European Conference on Pattern Languages of Programs.

Iivari, J. (2007). "A Paradigmatic Analysis of Information Systems as a Design Science," *Scandanavian Journal of Information Systems,* Vol. 19, No. 2.

Khazanchi, D. (1996). "A Philosophical Framework for the Validation of Information Systems Concepts," *Americas Conference on Information Systems*, Phoenix, AZ, 755-757.

Khazanchi, D. and Zigurs, I. (2005). *Patterns of Effective Management of Virtual Projects: An Exploratory Study*. Newtown Square, PA: Project Management Institute.

Khazanchi, D. and Zigurs, I. (2006). "Patterns for Effective Management of Virtual Projects: Theory and Evidence," *International Journal of e-Collaboration,* Vol. 2, No. 3, 25-48.

Khazanchi, D. and Zigurs, I. (2007). A Systematic Method for Discovering Effective Patterns of Virtual Project Management. Working paper available at SSRN: http://ssrn.com/abstract=1012080.

Klein, H. K. and Myers, M. D. (1999). "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems," *MIS Quarterly,* Vol. 23, No. 1, 67-93.

Lea, D. (1994). "Christopher Alexander: An introduction for object-oriented designers," *ACM Software Engineering Notes,* Vol. 19, No. 1.

March, S. T. and Smith, G. F. (1995). "Design and Natural Science Research on Information Technology," *Decision Support Systems,* Vol. 15, No., 251-266.

Maznevski, M. L. and Chudoba, K. M. (2000). "Bridging space over time: Global virtual team dynamics and effectiveness," *Organization Science,* Vol. 11, No. 5, 473.

Meszaros, G. and Doble, J. (1996). "A Pattern Language for Pattern Writing," *Pattern Languages of Programs Conference*, Allerton Park, IL.

Nunamaker, J. F., Chen, M., and Purdin, T. D. M. (1990-1991). "Systems Development in Information Systems Research," *Journal of Management Information Systems,* Vol. 7, No. 3, 89-106.

Prechelt, L., Unger-Lamprecht, B., Philippsen, M., and Tichy, W. F. (2002). "Two Controlled Experiments Assessing the Usefulness of Design Pattern Documentation in Program Maintenance," *IEEE Transactions on Software Engeineering,* Vol. 28, No. 6, 595-606.

Pries-Heje, J., Baskerville, R., and Venable, J. (2008). "Strategies for Design Science Research Evaluation," *European Conference on Information Systems*, Galway, Ireland.

Robson, C. (2002). *Real World Research,* (2nd ed.). Malden, MA: Blackwell Publishers.

Rossiter, J. R. (2002). "The C-OAR-SE Procedure for Scale Development in Marketing," *International Journal of Research in Marketing,* Vol. 19, No., 305-335.

Schuemmer, T. (2003). "Evolving a Group Pattern Language," *Fifth International Workshop on Collaborative Editing*, Helsinki, Finland.

Sproull, N. L. (1995). *Handbook of research methods: a guide for practitioners and students in the social sciences,* (2nd ed.). London: The Scarecrow Press, Inc.

Straub, D. W. (1989). "Validating Instruments in MIS Research," *MIS Quarterly,* Vol., No., 146-169.

Straub, D. W., Boudreau, M.-C., and Gefen, D. (2004). "Validation Guidelines for IS Positivist Research," *Communications of the Association for Information Systems,* Vol. 13, No. Article 24, 380-427.

Tidwell, J. (2005). *Designing Interfaces: Patterns for Effective Interaction Design*: O'Reilly Media.

Trochim, W. (2000). *The Research Methods Knowledge Base,* (2 ed.). Cincinnati, OH: Atomic Dog Publishing.

Vaishnavi, V. and Kuechler, W. (2008). Design Research in Information Systems [Electronic Version]. *AISnet*. Retrieved 2008, July 16, from http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=279

Vaishnavi, V. K. and Kuechler, W. (2007). *Improving and Innovating Information & Communication Technology: Design Science Research Methods and Patterns*: Taylor & Francis.

Venable, J. (2006a). "A Framework for Design Science Research Activities," *Information Resource Management Association Conference*, Washington D.C.

Venable, J. (2006b). "The role of theory and theorizing in design science research," *1st International Conference on Design Science Research in Information Systems & Technology*, Claremont, CA.

Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. (1992). "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research,* Vol. 3, No. 1, 36-59.

Wania, C. E. and Atwood, M. E. (2009). "Pattern Languages in the Wild: Exploring Pattern Languages in the Laboratory and in the *Real World*," *Design Science Research in Information Systems and Technologies*, Philadelphia, PA.

Weick, K. E. (1989). "Theory construction as disciplined imagination," *Academy of Management Review,* Vol. 14, No. 4.

Yoshioka, T., Herman, G., Yates, J., and Olikowski, W. (2001). "Genre Taxonomy: A Knowledge Repository of Communicative Actions," *ACM Transactions on Information Systems,* Vol. 19, No. 4, 431-456.