THE DESIGN THEORY NEXUS

By: Jan Pries-Heje
Roskilde University
Roskilde
DENMARK
janph@ruc.dk

Richard Baskerville
Georgia State University
Atlanta, GA 30303
U.S.A.
baskerville@acm.org

Abstract
Managers frequently face ill-structured or “wicked” problems. Such problems are characterized by a large degree of uncertainty with respect to how the problem should be approached and how to establish and evaluate the set of alternative solutions. A design theory nexus is a set of constructs and methods that enable the construction of models that connect numerous design theories with alternative solutions. It thereby offers a unique problem-solving approach that is particularly useful for addressing ill-structured or wicked problems. For each alternative solution in a design theory nexus one or more unique criteria are established to formulate a specific design theory. We develop a general method for constructing a design theory nexus and illustrate its utility using two field studies. One develops and applies an organizational change nexus. The other develops and applies a user involvement nexus. Each is a specific instantiation of the general design theory nexus constructs. Using these illustrations, we provide examples of how to evaluate such instantiations. We then discuss our findings as well as the validity of our approach. We conclude that the design theory nexus provides a viable conceptualization that enables the construction of effective problem-solving artifacts.

Keywords:

Introduction
The decision-making situations of real life involve two elements: factual and value (Simon 1960). These elements invoke two kinds of decision-making processes: structured and judgmental (Keen and Scott Morton 1978). Structured decisions are repetitive and routine. Unstructured decisions defy programmed decision making because they are novel, complex, or value-laden. Semi-structured decisions combine these two extremes: settings where the unstructured judgments of decision makers would be insufficient without support through more structured data modeling or information gathering.

Perhaps the apex of unstructured decision making occurs in the context of wicked problems. Such problems are poorly formulated, confusing, and permeated with conflicting values of many decision makers or other stakeholders. Because every outcome of the decision is obscure, the problem cannot be parted or solved piecemeal (Churchman 1967). Wicked problems share certain characteristics. For example, they can only be formulated in terms of a solution. Their solutions are value-laden, and cannot be denoted true or false, only good or bad. Their solution space is unbounded, and solutions are irreversible (Rittel and Webber 1973). Wicked problems cannot be approached, nor can alternatives be established or

---

1 Veda Storey was the accepting senior editor for this paper.
evaluated, without engaging considerable uncertainty. Yet social, environmental, and economic conditions are increasing the wicked problems facing organizational decision makers today (Courtney 2001).

The technical orientation of decision support systems makes them unsuitable for wicked problems (Mackenzie et al. 2006). For example, decision making requires one or more criteria. With only one criterion—price for an IS commodity, for example—it is easy to rank alternatives for selection, but two criteria mean determining the weighting for each one. With multiple criteria decision making (MCDM), the preferences of the decision maker become very important (e.g., how much is extra performance of a system worth). Further it also becomes important when the knowledge about preferences becomes explicit (Hwang and Masud 1979).

There is little practical use for interactive MCMD applications, perhaps because of their high complexity (Kaliszewski 2004). There are at least two other technical problems. For example, the multiple attribute utility theory (MAUT) (Belton and Stewart 2001; Keeney and Raiffa 1976, 2002) and the analytical hierarchy process (AHP) (Saaty 1988; Saaty and Vargas 1987) both use decision matrices where each decision alternative is assessed on each of the criteria. One problem is the assumption that a single number or an interval can be used to represent the preferences of a decision maker (Wang et al. 2007; Xu et al. 2006). The other problem is the assumption that the same set, tree, or hierarchy of criteria can be used to compare the decision alternatives. Many decisions in real-life organizations raise alternatives that are so essentially different that analytical comparisons invariably become biased because the analytical criteria will relate well to only a subset of the alternatives. This asymmetrical criteria problem is represented in Figure 1. The left bar graph represents a hypothetical set of four criteria and four options, where each option satisfies multiple criteria to a greater or lesser degree. The right bar graph represents a different hypothetical set of four criteria and four options. However, each option essentially satisfies its own criterion completely, while poorly addressing the criteria of other options.

This asymmetrical criteria problem is wicked because the solution introduces a singular criterion by which the problem is formulated, and decision becomes a value judgment about which problem criterion to address. It remains one of the most fundamental issues of ill-structured problems: a divergence between alternative formulations of the problem (Mitroff and Mason 1980). Decision support tools are still known today to be unsuitable for such wicked problems (Mackenzie et al. 2006). Examples of the asymmetrical criteria problem in information systems include the selection of ideal approaches from competing organizational change theories. Such theories involve vastly differing assumptions about organizational behavior, structure, interaction with technology, etc. The complexity of such selection decisions perplexes practitioners, and even senior academic researchers in the field faced with similar decisions in formulating research designs (Holmström and Truex 2001).

The decision is further complicated where decision makers concoct an “ideal” change method from a combination of features from multiple approaches. Such innovations raise issues of consistency and necessarily involving risky, irreversible experimentation in live, working organizations.

The research described in this paper uses design science and the concept of a theory nexus to distinguish logic, theories, and artifacts aimed at a general, operational decision tool that approaches the asymmetrical criteria problem.

Design Theory Nexus

There are differing opinions about what constitutes design theories for information technology artifacts. Walls et al. (1992) specify two major components of IT design theories: a product component and a development process component. Each draws upon kernel theories (usually taken from the natural or social sciences) in specifying prescriptive hypotheses that enable designers to evaluate whether the product and its development process satisfy the design theory. Goldkuhl (2004) specifies a need for multiple grounding of design theories in external theories, reference theories, value theories, etc. Markus et al. (2002) take a more practical view of design theories, using these theories to explain the means–ends relationship as a practical, prescriptively causal mechanism to justify design components.

These experts agree, however, that design research involves designs that are clearly driven by underlying theories. Theories are important elements when used as external, value, or kernel theories in IT artifact design for the purpose of insuring that work practices and IT remain correlated and synergistic.

We are concerned with the design of an artifact to improve the design of problem-solving approaches where a number of competing approaches exist. In particular, we address such settings where each theory has a distinctive design logic (i.e., responding to different parameters and constraints, focusing
on different utility functions, and operating with different
command variables). The approach we adopt is similar in
inspiration to Van Aken’s (2005) distinctions between the
algorithmic versus heuristic application of technological rules:
A technological rule is “a chunk of general knowledge linking
an intervention or artifact with an expected outcome or
performance in a certain field of application” (p. 23). The
algorithmic application of a technological rule is highly
deterministic, similar to a statistical determination. The heu-
ristic application offers the rule as a design exemplar for
which a practitioner would develop a special variant for a
specific case. However, in our case, practitioners must not
only apply heuristic rules, they must also choose from among
rather different kinds of technological rules.

We propose a design theory nexus to improve the idealized
design of problem-solving approaches where a number of
competing approaches exist, and where each theory has
distinctive design logic. A design theory nexus is a set of
constructs and methods that enable the construction of models
that connect numerous design theories with alternative
solutions.

Nexus (and its verb form necto) are Latin terms for binding,
fastening, and joining. Nexus is retained in English as a term
for “a bond, link or junction; a means of connexion between
things or parts” (OED 1989). Particularly important for its
usage in design science research, it also connotes “a con-
nected group or series” (Traupman 1966).

Carroll and Kellogg (1989) used the term nexus to ack-
nowledge the interactive nature of theory and design artifacts.
Using examples from human–computer interface, they explain
how the use of multiple psychological theories overdetermine
designed artifacts, and that the exact effects of an artifact in
relation to the theories underlying its design can only be
realized by experience with the artifact. A design theory
nexus extends the deductive view of the relationship between
to a reciprocal relation between
the articulation and rearticulation of theoretical claims and
iterations of design (Carroll and Kellogg 1989).

The most interesting theoretical contribution of a design
to a reciprocal relation between
the articulation and rearticulation of theoretical claims and
iterations of design (Carroll and Kellogg 1989).

March and Smith (1995) distinguish between design sciences
as involving building and evaluation while natural sciences
involve theorizing and justifying. Four design science
elements are

1. Constructs are concepts that characterize a phenomenon
2. Models describe tasks, situations, or artifacts
3. Methods show how to carry out activities toward a goal
4. Instantiations are physical implementations

We define five constructs: goals, environment, alternative de-
sign theories, theory nexus, and design solutions (see Figure 2).
Goals are “what the system is for” (Gregor and Jones 2007, p. 32), or its utility function (Simon 1996), or its meta-requirements (Walls et al. 1992, p. 40). Environment is a construct to denote contingencies that are outside the control of the people involved. Alternate design theories are each referring to design theories using the imperative logic of Simon (1996). The theory nexus is a connection point at which design theories bind with realities into a design solution. Finally, a design solution is the result constructed from highly dissimilar decision alternatives.

Indeed, in certain settings, the best alternative may have been to simply make no real selection at all, but to allow the organization to emerge organically.

Contingency theory arises from a broad array of studies, for example, as a model explaining the high failure rate for information systems in developing countries (Heeks 2002). The core idea in contingency theory is that organizations wanting to optimize performance need to adopt the structure that fits best with the situation they are in—the contingencies given them. Donaldson (2001) defines contingency theory: “At the most abstract level, the contingency approach says that the effect of one variable on another depends on some third variable” (p. 5).

Contingency models are advocated in difficult settings that involve ambiguous goals, multiple perspectives, and information that is susceptible to multiple interpretations, especially where interactive multi-person communication medium, such as face-to-face meetings, are impractical (Galegher and Kraut 1994). For example, contingencies such as communication support, process structuring, and information processing (Zigurs and Buckland 1998; Zigurs et al. 1999) condition task complexity at the group level in studies of task/technology fit (Van de Ven and Drazin 1985). Task/technology fit can have
Contingency theory has been criticized for having a too static view of the relationship between structure and technology (Barley 1990). It depends on shared characteristics of various organizational settings for determining the criteria by which one of an alternative set of problem-solving approaches may be evaluated. Contingency theory predicts or programs the selection by specifying these criteria. Operation of the theory means matching the characteristics of the potential host organization to the characteristics of the problem-solving approach using the criteria. Such criteria may be said to be symmetrical because they are similar from setting to setting and approach to approach. This operation breaks down when the problem-solving approaches or the problem settings themselves (or both) are so fundamentally different that their most prominent characteristics are no longer usefully comparable. In a worst-case scenario, the criteria for each approach and the criteria for each problem setting will be unique. This uniqueness prevents the pre-programming of practical selection criteria.

For example, this situation can arise when organizations are in a state of continual change, a condition known as emergence. Such situations have been recognized as hostile settings for mechanistic organizations and planned methodologies, and more conducive to organic organizations and agile approaches (Burns and Stalker 1961). As a result, contingency theory provides an avenue for selecting planned approaches, but is less effective in emergent organizations (Truex et al. 2000).

Method engineering provides an example of the degree to which contingency theory dominates the solution space for competing problem-solving approaches and the issues that arise. It provides frameworks and processes for assembling IS development methods from existing methodologies, inventories of method fragments, and innovations (Brinkkemper et al. 1996). With a focus on the formal models of computer-aided method engineering (Odell 1996), computer-based method engineering enables the rapid development of computer-aided systems analysis and software engineering (CASA/CASE) tools to uniquely match development settings. The concept of a method fragment or method chunk is one of the central method engineering premises (Rolland and Prakash 1996). A method fragment is any element of a methodology’s guidelines for its activities that is separated from the methodology. It is a concept, notation, tool, technique, etc. that is lifted from the framework of an overall methodology and interjected in a specific work practice (Baskerville and Stage 2001).

With an orientation toward formal modeling, fragment selection process is a straightforward engineering decision, taken in a technical rational debating process (cf. Harmsen et al. 1994; Oinas-Kukkonen 1996). For example, a technique–task matrix can be used to map techniques such as prototyping and role modeling onto project tasks such as code, model, plan resources, test, etc. (Henderson-Sellers 2003). The fragment selection process extends to consider the rationale underlying method use. This rationale is foundational for the use of evolutionary forms of method engineering. Such evolutionary forms permit instances of methods to be generated as business and technology changes (Rossi et al. 2004). Field studies have shown that the continuing integration of fragments permits situated method engineering as developers learn about this changing business and technological setting (Van Slooten 1996). It can be used to guide software process improvement settings to develop agile methods or object-oriented methods from method fragments (Henderson-Sellers and Serour 2005). Compound fragments can shape method components that express the transformation of particular artifacts along with the rationale for this transformation (Karlsson and Wistrand 2006). Componentization leads to method configuration, a form of meta-method and subdiscipline of method engineering that regards adjustments to configurable off-the-shelf methods (Karlsson and Ågerfalk 2004). Such method components can transform human-oriented conceptual models into system-oriented and machine-oriented design models (Wand 1996).

Although more sophisticated method structures (such as method components) have extended the original concept of method chunks, and method engineering has more sophisticated concepts for integrating social, pragmatic, and semiotic elements, method engineering necessarily operates as a contingency approach. Elements, although ever more sophisticated, are available for modification and integration based on preprogrammed characteristics. It accordingly depends on symmetrical criteria for selecting among method fragments, that is, settings in which the same criterion is used to adopt one method fragment while rejecting competing fragments. Consequently method engineering suffers from the same limitations as contingency theory in settings with asymmetrical criteria.

Within method engineering, this issue is denoted as a “multiperspective problem” (Nuseibeh et al. 1996, p. 268) arising from potentially conflicting notions of development strategies,
notations, etc. that inhabit the thinking of various systems
development stakeholders. Meta-methods, such as method
engineering, must capture and manage these conflicts. This
process yields technological artifacts into which the social
world has become embedded (Ågerfalk et al. 2006). This
integration also extends to the pragmatic and semiotic
elements of the practical settings (Goldkuhl and Ågerfalk
2004). But ultimately, symmetrical criteria are necessary to
guide the integration of these artifacts and elements. Baumoel
(2005) provides an example of the extensive research and
analysis necessary to construct such a criteria framework for
method engineering in selecting organizational change
approaches: a situated method construction approach. She
analyzed 89 cases and 23 change methodologies for the
purpose of developing an activity tree to guide the construct-
tion of change methods. By contrast, asymmetrical settings
are those in which the criterion for adopting one method
fragment, component, or element may be entirely different
from the criterion for adopting a competing method fragment,
component, or element. The theory nexus approach provides
an ideal setting for such asymmetrical settings.

Methodology: Defining Three Levels

Design involves planning for the construction of artifacts. While it is often contextualized in the fields of the humanities
(such as art, architecture, theater, etc.), its prominence in
fields such as economics, science, and engineering is growing
(Simon 1996). With this growing prominence, design for the
purpose of creating scientific and technical artifacts is
attracting increasing interest in the study of IT artifacts, an
interest simultaneously fueled by the need for more relevance
for economic studies of information technology (Orlikowski
and Iacono 2001).

Design science is a “science of the artificial” (Simon 1996)
that involves the scientific adaptation of means–ends
rationality to an environment. Designers, with ends in mind
(a goal), search for the means by which artifacts will achieve
those ends. Design science connects human aims to artifacts
that achieve those aims. Design science develops artifacts
that are acknowledged as subjects of natural law (Simon
1996, p. 3). Within the IS discipline, design science is most
directly concerned with the design, specification, and eval-
uation of technological artifacts.

At its foundations, design involves an imperative style of
logic, a search for alternatives, and the evaluation of design.
The logic of design is imperative because it is not a statement
of the way things “are,” but rather a statement of the way
things “should become.” Formal imperative logic is prob-
lematic (see Simon 1996, p. 115), so design research substi-
tutes an imperative adaptation of declarative logic. Once a
problem is expressed as a declarative paradigm, the design
process is one of searching through alternative values among
the logical terms in the paradigm until an alternative is
discovered that satisfies the goals of the design. Simon classi-
fied logical terms as command variables, fixed parameters,
constraints, and utility functions. Fixed parameters represent
the laws of the environment. Utility functions represent the
design goals. The values of command variables express
design alternatives. Constraints are a special form of param-
eter, one that may be selected according to the design goals.
Under this imperative paradigm, design becomes a search
process: Given the constraints and fixed parameters, find
values of the command variables that satisfy the utility
function (Simon 1996, p. 117).

Ultimately, the logic and the search should result in designed
artifacts that can be introduced into the information system
context. Examples of such artifacts include algorithms,
human–computer interfaces, system design methodologies,
and protocols (Vaishnavi and Kuechler 2004). Design valid-
ity is proven by an evaluation of these artifacts and their
achievement of the design goals (the satisfaction of the utility
function).

Walls et al. (1992, 2004) proposed a generalized design
theory that models the imperative logic paradigm, the search
for alternatives, and the evaluation of the artifact. Their
design theory has five components: meta-requirements, meta-
design, design method, kernel theories, and testable hypothe-
ses. Their model is important because it operates at a meta-
meta-level that provides a very general frame within which
classes of design approaches (meta-designs) can be defined
and described (see Table 1).

Gregor and Jones (2007) extend design theorizing into imple-
mentation, which is consistent with the concept of the artifact
as a theory nexus that involves rearticulation of theories in
experience. Their model defines eight components: (1) pur-
pose and scope, similar to meta-requirements, (2) constructs,
similar to parameters, variables, and constraints, (3) principles
of form and function, similar to utility functions, (4) artifact
mutability, or predicted state changes of the artifact, (5) test-
able propositions, similar to hypotheses, (6) justificatory
knowledge, similar to kernel theories, (7) principles of imple-
mmentation, similar to design method, and (8) expository
instantiation, which regards artifact implementation.

We operationalize Simon’s imperative logic, search, and eval-
uation at three levels, similar to Sprague’s (1993) framework
d of DSS technology. The three levels of this framework are
Table 1. Comparing Walls et al.’s Design Theory and Simon’s Imperative Logic

<table>
<thead>
<tr>
<th>Design Element</th>
<th>Design Theory Components</th>
<th>Description</th>
<th>Imperative Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Kernel Theories</td>
<td>Natural or social science theories governing design requirements</td>
<td>Fixed Parameters</td>
</tr>
<tr>
<td>Product</td>
<td>Meta-requirements</td>
<td>Applicable class of goals</td>
<td>Constraints and Utility Function</td>
</tr>
<tr>
<td>Product</td>
<td>Meta-design</td>
<td>Hypothetical class of artifacts meeting meta-requirements</td>
<td>Command Variables</td>
</tr>
<tr>
<td>Product</td>
<td>Product Hypotheses</td>
<td>For testing meta-design against meta-requirements</td>
<td>Design Product Evaluation</td>
</tr>
<tr>
<td>Process</td>
<td>Kernel Theories</td>
<td>Natural or social science theories governing design process</td>
<td>Search Process</td>
</tr>
<tr>
<td>Process</td>
<td>Design Method</td>
<td>Procedure for constructing the artifact</td>
<td>Search Process</td>
</tr>
<tr>
<td>Process</td>
<td>Process Hypotheses</td>
<td>For testing results against meta-design</td>
<td>Search Process Evaluation</td>
</tr>
</tbody>
</table>

Table 2. Three Operationalization Levels in Our Research the Nexus Design Research

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specific DSS</td>
<td>(1) Organizational Change Nexus instantiation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) User Involvement Nexus instantiation</td>
</tr>
<tr>
<td>2</td>
<td>DSS Generator</td>
<td>General method for constructing a Design Theory Nexus</td>
</tr>
<tr>
<td>3</td>
<td>Fundamental Elements</td>
<td>Design Theory Nexus</td>
</tr>
</tbody>
</table>

At the third (the highest) level, we develop the concept of a design theory nexus as an approach to a class of problems in which the design alternatives do not share similar criteria for their adoption (a design search across dissimilar or asymmetrical criteria). This framework allows us to instantiate a method for designing solution design approaches for active use in organizational settings (i.e., a general method for constructing a design theory nexus). This method comprises the second level in the framework and provides an approach to the specific class of problems requiring asymmetrical-criteria design searches. This second level artifact, method to construct (design theory) nexus instantiation, is our decision-system generator. Using this method, designers can search a design alternative space for ideal solutions, and even integrate alternatives to form uniquely ideal solutions. Out of this search, they can instantiate a design solution to operate in their organizational setting. This design solution corresponds to the first level in the framework, a specific decision system aimed at a real problem. We evaluate this approach using two real design settings, which leads to two different, specific implementations of these artifacts. Table 2 compares the operationalization of Sprague’s framework with our research.

General Method to Construct an Instance of a Design Theory Nexus

Design theories need principles of implementation to explain and guide the creation of the artifacts (Gregor and Jones 2007). We adapted the basic process involving (1) alternatives evaluation, (2) analysis, (3) design, and (4) implementation. We elaborated the design component into two distinct steps in order to distinguish between assertion development and assertion evaluation (see below). This model led to a five-step method for constructing an instance of a theory nexus artifact.

The first step in constructing a nexus instantiation is an analysis of the different approaches available in the given area of innovation. This analysis required a survey of existing
literature and findings. Literature about these kinds of problem settings may be overwhelmingly large, inconclusive, or both. In such cases, developers can supplement the literature survey with one or more additional analytic techniques such as a search conference or the qualitative interview study described in the cases below.

The second step involves analyzing the alternative approaches discovered in the first step, carefully mapping out the ideal conditions under which each approach has the highest utility. It may be the case that the approaches are analytically unequal such that hardly any conditions are equivalent for any pairing of the approaches. For example, this was the case in our analysis of organizational change approaches. In other settings many conditions may be equivalent, such as in the case in our user involvement nexus instantiation.

The third step shifts the process from an analytic to a constructive design of an artifact that can be used to indicate whether the conditions identified in step 2 can be found in an actual problem setting. The existing conditions are formulated as assertions.

The fourth step is to design and develop a decision-making process for evaluation of the formulated assertions. Depending on the area of innovation, there may be more or less facilitation required in this decision-making process. For the organizational change nexus instantiation, we found it ideal for the management group in each organization to evaluate and agree upon the assertions because management plays a key role in organizational change. A facilitated process was necessary. For the user involvement nexus instantiation, the problem area was closer to project managers working early in an IT development project. It was ideal for these project managers to evaluate the assertions because the conditions were more equivalent, the process was simpler, and less facilitation was required.

Finally the approaches, conditions, assertions, and the process are integrated into a tool (an artifact) to support the evaluation. In both cases, spreadsheet software proved adequate for developing the tool. The purpose of the tool is to calculate the “fit” for each approach in a given situation. We ultimately calculated fit in both instantiation settings using a scale from 0 to 100 percent, although other scales, such as a 1 to 5 maturity scale, could be more suitable for other problem settings.

Design Theory Nexus Instantiations

The design theory nexus originated in a project aimed at the discovery of ideal approaches to organizational change for specific organizational settings. It has subsequently been used for resolving a similar approach discovery process for improving user involvement in IS development. Consequently we will describe these two projects in the following sections.

Organizational Change Nexus Instantiation

The study of change is important in the management discipline. Academic and practitioner contributions to organizational change have been built on empirical work in a wide variety of organizations and from such different perspectives as psychology, sociology, and business. Examples of this work include descriptive accounts of change (e.g., Orlikowski 1993), normative models to guide the change process (e.g., Kotter 1996), theoretical models for understanding and analyzing change (e.g., Markus and Robey 1988), typologies of different approaches to organizational change (e.g., Huy 2001), and empirical studies of the success or failure of change (e.g., Hong and Kim 2002; Stelzer and Mellis 1998).

With the increasing interdependence of IT and organizational structure, behavior, and strategy, change in organizations inevitably involves correlated change in organizational IT. Regardless of the causal direction (Markus and Robey 1988), the benefits of the coordinated design of both work practices and related IT artifacts are long established (Ehn 1988; Mumford 1983). Theories of organizational change are candidates for kernel theories in IT artifact design for the purpose of insuring that work practices and IT remain correlated and synergistic.

There are many different foundational theories for organizational change. The process by which an organization selects the appropriate approach to organizational change is often ad hoc. Each approach has its advocates and adherents, and there is little comparative research for choosing among such approaches. These approaches are so varied that comparisons are usually drawn between only two (e.g., Beer and Nohria 2000), three, or four alternatives (e.g., Huy 2001).

The clients in this case were from four larger organizations in the financial sector. The case as described here was part of a larger, three-year research project operated by a research consortium of universities and client organizations with a total project budget of more than $5 million. The focus in the research consortium was improving the ability to improve.

An early experiment in one of the client organizations involved a team of three researchers and five practitioners in developing a very simple framework based on Beer and Nohria’s (2000) theories E and O and the four ideal types of
intervention used by Huy (2001). This experience revealed no effective way of establishing the situated conditions in terms of the various theories, and therefore no way of establishing causal relationships between the conditions in the organization and the theories. An early, very pragmatic research question arose: How do we prescribe one of the myriad change approaches to recommend in an organizational setting?

We developed the fundamentals of the design theory described earlier and began an implementation using the design theory nexus. Following the five steps of the method to construct the organizational change nexus instantiation, we analyzed the different change approaches that participants in the project had identified.

Step two involves analyzing the alternative approaches identified in the first step. From search conferences, we were able to identify a number of high-level overall approaches. We analyzed each approach in order to determine its distinguishing characteristics and to relate it to theories described in the literature. In particular, we focused on the essential goals of each change strategy (the ends) and on the essential processes (the means). In this way, we refined the approaches into 10 prominent organizational change strategies, summarized in Table 3. These strategies represent the design theories used as inputs to the nexus artifact. We then created a tool to guide change managers in evaluating which of the 10 change strategies would be most appropriate in their organizational setting.

**Constructing the Organizational Change Nexus Instantiation**

Step three shifted to the constructive design of an artifact that would guide designers in evaluating and choosing which of the 10 design theories, or what combinations of the 10 design theories, would be most appropriate. This design theory nexus provided the analytical tool to bind the various organizational change theories to actual organizational settings.

For each of the 10 organizational change theories (our implementation-level design theories) in Table 3, we formulated a number of assertions that were based on the prominent characteristics of each design theory as expressed in the literature referenced above. For example for the change approach called commanding, we formulated the following assertions (based on Huy 2001):

1. Right now we need change to happen fast.
2. It is primarily organizational structures that need to be changed.

3. In the past we have had successes in requiring or dictating change.

As another example, for the change approach called optionality, we formulated the assertions (based on Rogers 2003):

1. Our employees are self-aware and always have an opinion.
2. We have very knowledgeable employees that know their areas well.
3. There are vast differences between the tasks of different employees.

All of the assertions were assembled into a query form implemented as a simple application in a common spreadsheet package shown in Figure 3.

This form embodies a design theory as statements representing the conditions of the organization, the employees, the change ahead, and the current use of metrics. These conditions can be compared to the conditions for each of the 10 change approaches (Table 3) and the fit can be defined on a scale from 0 to 100 percent. The fit of these conditions can then be measured by the degree to which these conditions are present in the organization based on the assertions in the form.

Following this scoring method, the fit for each of the 10 change approaches can be determined. We let 50 percent represent an indeterminate item. Thus a fit calculated at 30 percent means that the related approach doesn’t fit at all (it is unfit). We let 100 percent indicate a complete fit of all conditions. (It is unlikely that any approach will achieve 100 percent in any actual situation).

Step four is to design and develop a decision-making process for evaluation of the formulated assertions in the field. A change strategy is heavily influenced by the vision or goals for the organization, as well as by whatever pertinent issues are present in the organizational culture. These issues will determine whether change strategies succeed or fail, and such issues belong to the top level of the organization.

We asked the management group in the IT division of the company to fill out forms similar to the one shown in Figure 3. The managers worked individually at first, and afterward we facilitated a discussion of any major differences brought forth in the individual assessments. For example, if one manager responded “agree” to the assertion “In the past we have had success in requiring or dictating change,” whereas another manager says “partly disagree,” then we would bring out that discrepancy in the discussion and use it as a basis for fostering fresh agreement within the IT management group.
Table 3. An Overview of the Ten Organizational Change Strategies (Design Theories)

<table>
<thead>
<tr>
<th>Name of Approach</th>
<th>Approach definition</th>
<th>Conditions</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commanding</td>
<td>Change is driven and dictated by (top) management. Management takes on the roles as owner, sponsor and change agents.</td>
<td>Where formal structures need change. Where change is needed fast.</td>
<td>Huy (2001), specifically the approach that is called commanding. The design and positioning schools as described by Mintzberg et al. (2002).</td>
</tr>
<tr>
<td>Employee driven</td>
<td>Change is driven from the bottom of the organizational hierarchy when needs for change arise among employees.</td>
<td>Where the need for change arises among the employees. Where there is no need for a standardized approach; the result is more important than the process. Where an open management style that will allow change to arise from the bottom.</td>
<td>Andersen et al. (2001) on the grassroots approach. Kensing (2003) and Kensing and Blomberg (1998) on participatory design.</td>
</tr>
<tr>
<td>Exploration</td>
<td>Change is driven by the need for flexibility, agility, or a need to explore new markets, technology or customer groups.</td>
<td>Where dynamic and complex surroundings makes it important to explore</td>
<td>Exploration (Benner and Tushman 2003), or the organizational structure called adhocracy (Mintzberg 1983).</td>
</tr>
<tr>
<td>Learning driven</td>
<td>Change is driven by a focus on organizational learning, individual learning and what creates new attitudes and behavior.</td>
<td>Where there is a need for change in attitudes and/or behavior. Where the organization is talented in learning. Where relationships between means and goals are unclear.</td>
<td>Huy (2001), specifically the approach called teaching. Also the learning organization (Senge 1990).</td>
</tr>
<tr>
<td>Metrics driven</td>
<td>Change is driven by metrics and measurements.</td>
<td>Where there are relatively stable surroundings so measurements from the past can be used to decide the future. Where the result of change is measurable.</td>
<td>Total quality management thinking (Oakland 2003). Six Sigma thinking (Pande and Holpp 2000).</td>
</tr>
<tr>
<td>Optionality</td>
<td>Change is driven by the motivation and need of the individual. It is to a large degree optional whether the individual takes the innovation into use</td>
<td>Where target group is very diverse and has large individual differences. Where individuals that should (could) change are highly educated, very knowledgeable and self-aware.</td>
<td>Rogers (2003) studied groups that took innovations into use voluntarily. Many of the models and techniques in Rogers are valid for this change approach.</td>
</tr>
<tr>
<td>Production organized</td>
<td>Change is driven by the need for optimization and/or cost reduction.</td>
<td>Where you have relatively stable surroundings. Where you have many homogeneous resources and workflows.</td>
<td>Scientific management, (Benner and Tushman 2003), Huy (2001), specifically the approach called engineering.</td>
</tr>
<tr>
<td>Reengineering</td>
<td>Change is driven by fundamentally rethinking and redesigning business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service and speed.</td>
<td>Where a need exists for major change. For example when organization has ground to a halt. Where nothing new happens. Where decisions are made but not carried out. Where a crisis is eminence.</td>
<td>Bashein et al. (1994), Boudreau and Robey (1996), Davenport (1993), Hammer (1990), Hammer and Champy (1993), King (1994), Malhotra (1998), Willcocks et al. (1997).</td>
</tr>
<tr>
<td>Socializing</td>
<td>Change in organizational capabilities is driven by working with social relationships. Diffusion of innovations happens through personal contacts rather than through plans and dictates.</td>
<td>Where organizational skills and capabilities needs to be developed. Where no unhealthy power struggles occur (so people can talk). Where employees that can be exemplars are available.</td>
<td>Cohen et al. (1972) and Huy (2001), specifically the approach called socializing.</td>
</tr>
<tr>
<td>Specialist driven</td>
<td>Change is driven by specialists, either with professional, technical, or domain knowledge. Examples are a method or architecture function.</td>
<td>Where work has vast complexity and variety so there really is a need for special knowledge. Where there is access to necessary specialists, eventually by in-sourcing them.</td>
<td>Ciborra (2000), Mintzberg (1983), especially professional bureaucracy, Simon (1973, 1983), Woods (1988), Woods and Hollnagel (1987).</td>
</tr>
<tr>
<td>Assertions/Statements</td>
<td>Agree</td>
<td>Partly agree</td>
<td>Neither nor</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------</td>
<td>-------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>The company and its context</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The organizational context is very dynamic and demands frequent change.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Too little is happening here - we have ground to a halt.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The organization is doing well and making money.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordination across the organization is lacking.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The work here is quite complex and depends on specialized knowledge.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The knowledge and skill we have could be used more optimally, through economies-of-scale and uniform processes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The employees</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The best ideas for change come from the bottom of the organization.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our employees are self-aware and always have an opinion.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>We have very knowledgeable employees who know their areas well.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The actions of different employees can vary widely.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>We depend on knowledge and specialists from outside.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizational change often happens without our contributing to it, but with our consent.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>We have unhealthy power struggles and signs of bad chemistry between people.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change in the organization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>We make change often.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The changes we initiate always succeed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right now we need change to happen fast.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right now there is much disagreement about what needs to be changed and what directions we should take.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primarily, organizational structures need to be changed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primarily, complex workflows need to be changed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primarily, attitudes and social relations need to be changed in the future.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up until now it has been the work with social relations that created change.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In the past we have had success in requiring or dictating change.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The results of change are much more important than the change process.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A specific (and separate) part of our organization takes care of exploring new things.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Experience and “best practices” are always handed down to new employees and new projects.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metrics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>We have a metrics program today—and we use the results.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is entirely possible to measure the outcome or result of change.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. The Form Used to Measure Actual Conditions in an Organization
As another example, if one person agrees that “Right now we need change to happen fast” and four persons partly agree, the partly-agree majority yields a calculation of 75 percent (with agree being 100 percent). For the second question, “It is primarily organizational structures that need to be changed,” if two people answer “agree” and three people answer “partly disagree,” we would initiate a facilitated discussion. When there was more than one field (in Figure 3) separating the answers, then managers should explain to each other why they answered as they did. In the discussions that follow, their reasoning leads to agreement. In this example, there was underlying disagreement about organizational structure. But after discussing their disagreement, all five decided on “partly agree.” This result is then used to calculate the fit as being 75 percent for this question in relation to commanding. The third question, “In the past we have had successes in requiring or dictating change,” was unanimously answered “disagree.” Disagree equals 0 percent fit. Finally, we summed up the fit for commanding as 75% + 75% + 0% divided by 3 = 50% fit.

The goal in this process is to determine a degree of fit of all 10 change approaches rather than to calculate a single ideal approach. Because real settings are complex, we allow the possibility (indeed the likelihood) that the ideal approach will involve a combination of features from two or more of the best-fitting approaches. In this way, our allowance of multiple, integrated change approaches is an extension of Van Aken’s (2005) notion that technological rules can sometimes best serve as heuristic exemplars in practice. The exact set of features to be included in this combination must be deferred to the change managers in the setting.

Implementation and Evaluation

Step five draws the approaches, conditions, assertions, and the process into the design of a tool (part of the implementation artifact) to support the evaluation. Our goal indicates that the justification and evaluation of the resulting artifacts should focus on whether the artifact operates in the problem setting. We use this simple operational goal because currently there are no other alternatives for this class of problems. Without competing solutions, making a better or worse performance comparison is not possible. Therefore, a nexus instantiation that can be shown to operate satisfactorily in helping the people to select alternative approaches where asymmetrical criteria prevail will be an improvement.

The design theory nexus is a human–machine problem-solving system involving methods as well as an IT artifact. Implementing and evaluating design solutions in a space where none currently operate might be best described as an exploratory form of design research. This exploratory usage is framed in a social-organizational setting highly characterized by change. Hevner et al.’s (2004, p. 88) guideline concerning design research rigor recommends that “the principle aim is to determine how well the artifact works, not to theorize about or prove anything about why the artifact works.”

Therefore, to evaluate the organizational change nexus instantiation empirically, we used the tool in analyzing two real organizations that were confronting change projects. Working through the research consortium, the four client companies involved in the consortium had agreed to participate. We first used an early form of the organizational change nexus in Danske Bank in April–May 2004. This first project failed because it was too complicated to establish causal relationships directly between contingencies and change approaches.

Based on this experience, we turned the solution upside down, developing the set of 10 change approaches using search conferences. For each change approach, we asked, under what circumstances would we recommend this approach?

In February 2005, we evaluated the first version of the design theory nexus artifact presented in this paper. The evaluation took place in PBS (Payment Business Services), a company that specializes in electronic payment services. PBS employs 750 people, of which half are working within the IT division. They develop and operate solutions for payment systems and are a leading supplier of such solutions and related services to banks, private associations, and public institutions, processing 1.3 billion transactions last year. A second evaluation—using for the first time exactly the design theory nexus presented in this paper—took place in ATP in September 2005. ATP provides solutions for pension and insurance applications for organizations and individuals in the labor market. ATP is Denmark’s largest pension scheme and most of the country’s workforce has accrued rights to one of ATP’s pensions. ATP has more than a 1,000 employees, 350 of whom work with IT.

A third round of evaluations took place, in May 2006, in three organizations. Here we tried the nexus instantiation with individual people, namely the CIO from each of three organizations: Housing, Energy and Communication (real names anonymized at the request of the CIOs). In Table 4 we show the results from the five evaluations.

For each organization, we looked at the one or two approaches with the best fit. At PBS, the two top approaches surfaced with exactly the same fit. Accordingly, we decided...
Table 4. Results from Using the Organizational Change Theory Nexus in Five Organizations†

<table>
<thead>
<tr>
<th></th>
<th>PBS</th>
<th>ATP</th>
<th>Housing</th>
<th>Energy</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reengineering</td>
<td>31</td>
<td>28</td>
<td>30</td>
<td>40</td>
<td>28</td>
</tr>
<tr>
<td>Optionality</td>
<td>54</td>
<td>*71</td>
<td>*92</td>
<td>*92</td>
<td>50</td>
</tr>
<tr>
<td>Socialization</td>
<td>*60</td>
<td>59</td>
<td>82</td>
<td>28</td>
<td>63</td>
</tr>
<tr>
<td>Specialist driven</td>
<td>37</td>
<td>56</td>
<td>*96</td>
<td>58</td>
<td>63</td>
</tr>
<tr>
<td>Exploration</td>
<td>35</td>
<td>29</td>
<td>21</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Commanding</td>
<td>34</td>
<td>*65</td>
<td>8</td>
<td>*63</td>
<td>75</td>
</tr>
<tr>
<td>Employee driven</td>
<td>55</td>
<td>18</td>
<td>23</td>
<td>60</td>
<td>63</td>
</tr>
<tr>
<td>Learning driven</td>
<td>*60</td>
<td>34</td>
<td>69</td>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>Metrics driven</td>
<td>42</td>
<td>40</td>
<td>58</td>
<td>23</td>
<td>*73</td>
</tr>
<tr>
<td>Production organized</td>
<td>56</td>
<td>58</td>
<td>63</td>
<td>50</td>
<td>*75</td>
</tr>
</tbody>
</table>

†Numbers are fit as measured between 0 and 100 percent. For each company, the two best fitting approaches are in bold and with a star in front.

to recommend a combination of these two approaches. We could have recommended three or four approaches but we believe the complexity of combining more than two approaches would confuse rather than help the organization. In this way, the organizational change nexus instantiation guided PBS to a recommendation for a combination of the change approaches socialization and learning driven. The nexus instantiation helped the managers analyze the change approaches in relation to their organizational vision and setting. Van Aken’s (2004, p. 227) notion of a “design exemplar” applies. The exemplar can be only a general outline that must be translated into proper form for the exact problem at hand.

The CIO (manager of the IT division) evaluated the results positively and enthusiastically. He called the results a major “Aha!” experience, and compared it to the wearisome exchanges with previous consultants who seemed to expect him to “run around with a box of matches” to establish a burning platform (referred to in phase 1 of Kotter 1996). A month after the evaluation, PBS began implementation of the organizational change strategy developed with its basis on the organizational change nexus instantiation. They aimed to achieve their strategic vision over the next 2 to 3 years. In terms of the development of a useful change strategy, this second application of the organizational change nexus instantiation may be regarded as a complete success.

At ATP we discussed whether we should just recommend one approach (the best at 65 percent of course), but based on the success experience at PBS, where we recommended two, we decided to similarly combine the top two approaches from the evaluation at ATP. The combination of optionality and commanding approaches was surprising and led to a discussion among the ATP managers as to whether these two approaches would fit together as a combination. Ultimately, ATP managers agreed that these could be combined by enabling managers to choose multiple change projects and driving each initiative separately, using the optionality approach in some and the commanding approach in others. In other words, managers select a few (two or three) initiatives where they personally drive the change (commanding) while at the same time make it clear that other initiatives (many, in fact) must be driven by the individuals’ or groups’ need and motivation (optionality). In this final evaluation by ATP, the organizational change nexus instantiation was also considered a success, in that it enabled ATP managers to analyze and understand complicated and alternative approaches to organizational change. They were able to confidently develop decisions about organizational change strategies relative to the vision and setting at ATP. However, since ATP had formulated a rather bold strategic vision, the present study of change approaches and its evaluation is too recent to provide indications about the actual impact of the results in driving this organizational change to success.

In the three companies using the organizational change theory nexus most recently, we conducted follow up interviews with their CIOs. Overall, they stated high levels of satisfaction, for example, on a scale from 1 to 5 (with 5 being best), all evaluated the experience of using the nexus artifact at 4 or 5.

**Evaluation Results**

The organizational change nexus instantiation is an artifact that provides an analytical tool to help organizational change managers formulate organizational change strategy. The design theory used in designing this analytical tool is an example...
of a design theory nexus. It draws from multiple sources of external or kernel theories, including those taken from 10 distinctly different change approaches. Analyzed together with the organizational environment and vision, these theories are rearticulated into specific change strategies for the setting as shown in Figure 4.

What constitutes success in using the organizational change nexus instantiation? We operationalized this measure based on user retrospection and the binary decision by the organization to implement the results (or not). These are simple measures of user satisfaction and impact. Overall, the design theory nexus is a decision making aid. Like any other information tool, there can be several alternative fundamental measures of success, such as quality, use, user satisfaction, and impact (DeLone and McLean 1992, 2003; Garrity and Sanders 1998). We could operationalize these in our case as increased speed of decision making, increased agreement on decisions, lower or higher satisfaction with the decision, or even the ability to arrive at a decision at all, and ultimately better decisions—for example, measured by better outcomes from implementing the decision.

Measuring better outcome would require a longitudinal study. Both PBS and ATP, for example, had strategic visions reaching 2 to 3 years ahead in time. Real world situations typically have complexity too high for unequivocal measures of that type of success. Measuring increased speed would require that the decision, the horizon, and the scope of the decision are exactly the same to enable us to compare hours and calendar time used. However, organizational change is seldom discussed alone. In practice it is typically discussed together with a number of other issues on the agenda in the particular organization. Therefore this measure of success would quickly become distorted.

We settled for a simple, straightforward measure of success that combined the subjective satisfaction of the involved managers with a measure of whether the actions undertaken in the organization were in fact influenced by the evaluation. In all five cases, this turned out in favor of the organizational change nexus instantiation.

**User Involvement Nexus Instantiation**

User involvement can be defined as “participation in the system development process by potential users or their representatives” (Barki and Hartwick 1989, p. 53). Many have identified lack of fulfillment of user needs in IT projects as a major problem. Clavadetcher (1998, p. 30) summarized the problem, “Quite simply, the software we build does not meet our customers’ needs...we fail to adequately manage the software development process. User-developer communication breaks down.”
A recent analysis of failing public projects concluded that the involvement of users in failing projects was flawed in key ways (Bonnerup 2001). In some cases, too many users were involved. There were also examples of user–developer communication breakdowns mainly because the abstraction level of the communication became too high for the users.

Traditionally, user involvement has been found to be a major factor in systems’ success. This finding builds on theories of participative decision making (Barki and Hartwick 1994) and the user role in organizational change (Baroudi et al. 1986). There is not total agreement on the benefits of user involvement (Ives and Olson 1984). Cavaye (1995) studied the relationship between user involvement and success and found that the relationship was more complex than just more user involvement leading to more success. One explanation may be that user involvement varies within each stage of the development process (Newman and Noble 1990). Further, Robey and Farrow (1982) point out that participation without influence is unlikely to lead to success. Noyes et al. (1996) highlight the difficulty in deciding how to involve users, and where they are involved, the difficulty in using them to their full potential. Thus it seems that user involvement is very important but also very hard to do in practice.

As with organizational change, there are also many different fundamental theories for user involvement. Bedker et al. (2004) provide an excellent overview of a number of these techniques for user involvement. For example, user involvement through paper-based prototypes is recommended early in the development process in order to elicit requirements from the users. Another example is Saleem (1996), who recommends user involvement when task uncertainty is high.

The process by which an IT project selects the appropriate approach and time for user involvement is often ad hoc. We set out to develop a Nexus instantiation for user involvement.

**Field Study**

This study engaged three researchers: one tenured professor (an author of this paper) and two students writing a dissertation on user involvement. In order to build a design theory nexus instantiation for user involvement, we researched methods and techniques for user involvement. Again we followed the method construct as described above. As step one, we identified hundreds of research papers and books on this subject and developed a set of alternative methods and techniques for user involvement. Based on this set of techniques, we then conducted a field study in 10 companies using exploratory interviews followed by semi-structured interviews. We then initiated step two of the method construct, analyzing the alternative approaches discovered in the first step. The combined findings from our literature study and our field study indicate that there are three major influences on user involvement: complexity, resources and user identity.

**Complexity.** The complexity issue is characterized by six major factors that give rise to complexity. These include domain knowledge, task complexity, size, technical knowledge, perceived change, and the type of system.

The first factor leading to complexity is the degree of knowledge held by the developers in the domain in which the users work. Developers need knowledge about the existing working styles in order to develop the right system for future work. Lack of domain knowledge among developers is one of the three key problems of systems development when the system is large (Curtis et al. 1988). Where developers lack knowledge of the users’ work, they need to observe and experience the users while they do their work (Kensing and Munk-Madsen 1993). A high degree of user involvement is needed where developers lack domain knowledge (Saleem 1996), and this involvement is less urgent where developers have strong domain knowledge. Cavaye (1995) notes that user involvement is less urgent if user requirements are well-known.

The second factor is task complexity. Where tasks are structured at the operational level, the demand for user involvement is minimal. Where tasks are unstructured and only described at the strategic level, the need for user involvement is urgent (Cavaye 1995). If a system is well structured and well defined, then it is not necessary to involve users for purposes of system quality (Ives and Olson 1984) but perhaps for system acceptance.

Size is the third factor we found that influences complexity. Size is one of the main influences both on system risk (Applegate et al. 1999) and complexity (Cavaye 1995). The distinction between large and small will, of course, vary from setting to setting; we asked participants in our field study how they would distinguish large from small systems.

The fourth influence factor on complexity was knowledge of the technology. Lack of technological knowledge increases complexity. This factor is a known area that increases risk (Applegate et al. 1999), and we know that there is a need for a balance between the complexity of the application and the complexity of the technology (Nicholas 1985). User involvement may be unsuitable when considerable technical expertise is needed (Ives and Olson 1984).
The fifth influence factor is perceived change for stakeholders. If change is perceived to be considerable, then there is an advantage in involving users (Cavaye 1995).

Finally the sixth influence factor is the type of system. For example, decision support systems involve more complex workflows and a higher degree of user involvement may be needed (Hawk and Dos Santos 1991). When developing standard applications, such as a payroll system, a small degree of user involvement may suffice (Saleem 1996).

Resources. Three major influence factors developed from the literature review that regarded resources. The first factor was management support, which will increase the prospect of user involvement (Cavaye 1995). Where senior managers are only moderately positive, or worse still, resistant, then risk increases considerably (Applegate et al. 1999). Management must support user involvement in both word and deed, along with commanding results from the involvement of users.

The second factor involves resources in the form of adequate budget and staff for the project (Cavaye 1995). When a project has limited resources, then users will be less involved simply because user involvement is expensive.

The third factor is time, especially with regard to whether the project has a hard or a soft deadline. Projects with hard deadlines can exclude effective engagement for user involvement (Wilson et al. 1997).

User Identity. The degree to which the users are personally known to the developers depends on the type of development. User identity is classified as named or nameless users (inspired by Grudin 1991). The most practical measure of this identity is whether the developers can name the users, or at least to obtain the names of the users in advance. In product development, developers do not know their users until the users buy the product. It is possible to identify potential users such as representative users of the last version of a product, but the complete set of users necessarily remain nameless until they acquire the product.

Eight Design Theories on User Involvement

Step three of our method construct focused on the constructive design of a nexus instantiation. The literature review and the field study led to the identification of eight prominent project types combining the three dimension (complex–not complex, many–few resources available, named–nameless users) two-by-two-by-two. Again, as with the organizational change approaches, we aimed at developing a design theory for each of the project types.

For each design theory we then analyzed the ideal setting to determine which of the techniques best fit the factor configuration. Following this analysis, we created a nexus instantiation that could guide system developers and project managers in deciding which of the eight design theories and what combination of techniques would be most appropriate.

In the process it became evident that the alternative approaches can be selected with symmetrical criteria. Project size, complexity, available resources, and user identity are criteria that are all shared among the competing approaches. While unpredicted, this symmetry made this case ideal for evaluating a design theory nexus instantiation in settings where criteria are symmetrical.

Constructing the User Involvement Nexus Instantiation

The fourth step in our method construct is to design and develop a decision-making process for evaluation using formulated assertions. In combining the influence factors of user involvement into the eight design theories, we formulated a number of assertions that would reveal whether the conditions were present in an organizational setting. These assertions were based on the prominent characteristics of each design theory as expressed in the literature and in the field study. All of the assertions were assembled into a query form using a spreadsheet package. The complete form is shown in Appendix A.

The design theory is represented in this form in that the statements represent an expression of the conditions of project complexity, resource situation, and development type in terms of named or nameless users. These conditions are derived from the conditions for each of the eight project types (Appendix A) and the fit as defined on a scale from 0 to 100 percent. For example, if we take an example set of conditions

1. When project is complex
2. When many resources available
3. When users can be named

The fit of these conditions can then be measured by the degree to which these conditions are present in the project. Some items fit with agreement (the better the fit, the more they agree) while others fit with disagreement (the better the fit, the more they disagree). Following this scoring method, the fit for each of the eight project types can be determined. The goal in this process is to determine a degree of fit of all eight project types rather than to calculate a single ideal type. Because real settings are complex, we allow the possibility (indeed the likelihood) that the best approach will involve a
### Implementation and Evaluation

Finally the fifth step in our method construct is where it all comes together in an artifact to support the evaluation. For the user involvement nexus instantiation we invited companies to a workshop in June 2006. People from 12 companies participated in a workshop that lasted 6 hours. All workshop participants derived project plans from the user involvement nexus instantiation. The workshop concluded with an evaluation by the participants. Overall, the design nexus instantiation is a tool to aid decision making. We sought a simple and quick evaluation of impact. The suitability of the user involvement nexus instantiation was measured according to a participant retrospective and the degree of their intention to implement the results in practice. These two measures provided simple, but effective metrics of the impact of the nexus instantiation on participants and their future behavior.

On a scale from 1 (worst) to 5 (best), we asked the workshop participants to evaluate the tool and the plan as well as satisfaction with the workshop as a whole. Table 5 shows both the form and the results of the evaluation.

<table>
<thead>
<tr>
<th>Evaluation of Nexus</th>
<th>😞</th>
<th>😞</th>
<th>😞</th>
<th>😞</th>
<th>Don’t know</th>
</tr>
</thead>
<tbody>
<tr>
<td>My expectations meet</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>4.0</td>
</tr>
<tr>
<td>Value of workshop</td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>3.7</td>
</tr>
<tr>
<td>The Nexus (spreadsheet)</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4.0</td>
</tr>
<tr>
<td>Derived schedule with milestones</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deviation: Your plan in relation to &quot;standard plan&quot;</th>
<th>Nothing</th>
<th>Little</th>
<th>Some</th>
<th>Partly</th>
<th>A lot</th>
<th>Don’t know</th>
</tr>
</thead>
<tbody>
<tr>
<td>To what degree will you expect to use the derived plan after today</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td></td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Three conclusions about the impact of the user involvement nexus instantiation are indicated from the evaluation. First, the workshop was deemed very valuable (3.7 average) but the user involvement nexus instantiation was even more valuable (4.0). Second the participants were quite satisfied with the user involvement plans that they derived when using the nexus instantiation (4 on a 1 to 5 scale). Third, and perhaps most importantly, nearly all participants deviated from the standard plan that was linked by the best fit to the project type, and nearly all participants expected to use the plans derived from the nexus instantiation, although 2 out of 12 only expected “little” use. Overall, the evaluation indicated that the user involvement nexus instantiation delivered the outcomes expected in accordance with its design logic.

### Discussion

The research above developed the design theory nexus concept at three levels as shown in Figure 5.

### Learning from the Instantiations

The importance of the operations using the both the organizational change nexus instantiation and the user involvement nexus instantiation lies in the implications for the design theory nexus’s scope. The first example, the organizational change nexus instantiation, provides evidence that nexus instantiations offer a general solution to the asymmetrical criteria problem. But does this instantiation offer an elaboration of existing approaches, such as contingency theory, or is it an alternative? Does the design theory nexus approach work for asymmetrical criteria problems only, or does it work for both symmetrical and asymmetrical problem settings?

An answer was given by the second instantiation, the user involvement nexus instantiation. In short, the answer was that it was possible to develop operational instantiations for both
kinds of settings. This work suggests that the design theory
nexus approach is more universal than previous approaches to
contingency theory, because it can operate in both sym-
metrical and asymmetrical settings. This wider scope means
that a project can start without knowing at the outset whether
the criteria will be symmetrical or asymmetrical. There is
evidence that a design theory nexus can be instantiated and
operate in both kinds of settings.

Learning from the Design Theory Nexus

Hevner et al. (2004) distinguish between two paradigms that
c characterize much of the research in the IS discipline:
behavioral science and design science. The behavioral
science paradigm seeks to develop and verify theories that
explain or predict human or organizational behavior, thus the
objective is problem understanding. The design science
paradigm seeks to extend the boundaries of human and
organizational capabilities by creating new and innovative
artifacts, thus the objective is problem solving.

Hevner et al. identify truth and utility as the core outcome of
the two different paradigms.

Behavioral science addresses research through the
development and justification of theories that explain
or predict phenomena related to the identified
business need. Design science addresses research
through the building and evaluation of artifacts
designed to meet the identified business need. The

goal of behavioral science research is truth. The
goal of design science research is utility (p. 79-80).

Hevner et al. presented a design science research framework.
At the core was a cycle between the build/develop and justify/
evaluate activities. In relation to Hevner et al., we developed
two different applications, namely one (instantiation) for
change approaches and one (instantiation) for user involve-
ment. Both of these applications are relevant for people, they
are useful in organizations, and they provide an IT artifact
(technology). In Figure 6, this is shown as the arrow from the
middle to the left.

However, our design theory nexus also contributed to the
knowledge base. As noted, we were looking at difficult deci-
dions where approaches such as MCDM (Belton and Stewart
2001), AHP (Saaty 1988), and MAUT (Keeney and Raiffa
2002) have been tried. We have added a design theory nexus
that can be used for highly dissimilar approaches. We have
also discussed how existing theories on contingency theory
and method engineering are related to the nexus. Further we
have discussed existing design theory (Hevner et al. 2004;
March and Smith 1995; Simon 1996). In Figure 6, our contri-
bution to the existing knowledge base is shown as an arrow
from the middle to the right.

Coping with Uncertainty

A design theory nexus copes with uncertainty rather than
manages it. It thrives in high-change environments because
Build

Evaluate

IS managers facing highly dissimilar alternative approaches to decisions

Business needs

Applicable knowledge

Knowledge Base
1. MCDM, AHP, MAUT etc.
2. Contingency Theory
3. Method Engineering
4. Existing Design Theory

Applications

Additions:

Change Approaches
User Involvement

Design Theory Nexus for highly dissimilar approaches

Figure 6. The Contribution in Form of Applications and Additions to Knowledge Base

the entire theoretical framework for determining action is open to negotiation. New reference theories can be added to the Nexus, theories whose selection might be based on entirely different criteria, without changing the overall nexus framework. This aspect of the design theory nexus is best illustrated by the case of the organizational change nexus instantiation, where there were fundamental differences between the criteria under which a particular theory would be selected as a driver.

A design theory nexus differs from contingency theory in the way that it promotes the integration of appropriate aspects of different alternative theories that might be suitable for the situation. The design theory nexus together with the method-to-construct may surface multiple alternatives, with indications of the degree of fit for the situation. Alternatives with similar fit (although perhaps based on differing assumptions) can be examined more closely for the development of hybrid approaches that combine features of multiple alternatives. This aspect of the design theory nexus is well illustrated by both the organizational change and the user involvement nexus instantiations.

In these ways, a design theory nexus has promise of helping resolve the primary limitation of contingency approaches. This limitation regards the difficulty in estimating precisely the factors involved in a contingency model (often called the interaction terms because it is the interaction of the factors that are interesting in a given situation). These interaction terms grow quickly irrelevant, “less than one-quarter of the interaction terms investigated...have been found to be significant” (Chin et al. 2003, p. 212). As a result of such flaws, the usefulness of contingency models is open to question (Weill and Olson 1989). Indeed, empirical research around contingency theory continues to provide inconsistent results (Cavaye 1995).

A design theory nexus operates well with both asymmetrical criteria and symmetrical criteria. This feature opens the possibility for the use of a design theory nexus within future method engineering systems as well as less formalized development settings.

**Design Theory Productions**

This study helps illuminate the process by which the design theory approach can generate theory. Existing theory binds with experience to generate a modified, reoriented, or entirely new theory as an emergent byproduct of the process of design. Design research is a setting in which theory and experience are engaged in generating new artifacts intended to change social and/or physical reality in purposeful ways. It is not so much a logical formulation process as it is an epiphany that follows a clash of old ideas in new situations.

Theory development in design research centralizes the designed artifact as a means of changing social and/or physical reality. This design artifact is an embodiment of the design theory, and its operating influence on its setting is the
phenomenon of interest. At the general design theory nexus level, the theory arises as a consequence of the inability of contingency theory to provide ineffective selection mechanisms for settings where asymmetric criteria dominate. This clash between theory and experience provides the kind of noisy setting where more satisfactory designs arise. In the case of the organizational change nexus instantiation, the theory arises in the potential clash between organizational change theories and the organizational reality. The outcome is the set of organizational change theories that may be integrated together to develop a unique organizational change theory for the exact setting at hand. Such a unique construction results from the process in which various organizational change theories bind with each other and the social/physical reality.

Validity of Approach

Hevner et al. (2004, p. 83) proposed a view for excellence in design science research in the form of seven guidelines that are useful for understanding, executing and evaluating design science.

1. Must produce a viable artifact.
2. Produces technology-based solutions to relevant business problems.
3. Evaluation that demonstrates utility, quality, and efficacy.
4. Research contribution of the design artifact, foundations, or methodologies.
5. Rigor in construction and evaluation method.
6. A problem-situated means-ends search for an effective artifact.
7. Communication to both technical and managerial audiences.

In relation to these particular guidelines, the design theory nexus is a designed artifact in the form of constructs, a model, a method-to-construct and two examples of instantiations (guideline 1). It provides relevance to an important business problem, namely how to make decisions as an IS manager when facing a problem setting with many competing, but highly dissimilar, alternative approaches, by helping resolve a multiplicity of alternative approaches into a more cohesive direction (guideline 2). The utility and efficacy of the design theory nexus was evaluated in two projects in which a method-to-construct was applied and two artifacts (instantiations) developed. These provided observational evidence from field studies to indicate that a nexus instantiation was operational and valued in practice (guideline 3). The organizational change nexus instantiation provided practical contributions as an artifact that helped to solve a complex problem common to many organizations, and it provided a research contribution by extending the concept of a design theory nexus beyond the HCI discipline (guideline 4). Our study did not adopt a strictly scientific form of design science, and it did not adopt tightly structured deductive design theory and hypotheses. The approach adopted an interpretive social research stance, one widely accepted as a rigorous form of research (guideline 5). The participative nature of the approach provided the mechanism for searching and utilizing the available means to reach the desired ends (guideline 6). Finally the studies provided evidence that the design theory nexus can be presented to both a technology-oriented and a management-oriented audience (guideline 7). For example, a number of change agents were present in the three cases where the organizational change instantiation of the nexus artifact were evaluated. The evaluation showed that management in the organizations was more interested in the overall change strategy.

Conclusion and Future Research

A common but intractable problem faced by many managers involves the selection of an ideal approach to a problem setting in the context of many competing, but highly dissimilar, alternative approaches. In particular, we have focused on settings where the available alternatives are different in essential ways. These essential differences arise because the competing approaches are based on theories with vastly differing assumptions about organizational behavior, structure, interaction with technology, etc. These problem settings lack comparative research that helps select the most ideal design theory for the precise problem setting at hand because the theoretical basis is different. Unbiased analytical comparisons of such approaches are impossible because the analytical criteria will relate well to only a subset of the alternatives.

We used concepts from design research to develop a generalized approach to these kinds of problem settings. Because the competing alternatives were related to competing theories, our approach regarded the alternatives as competing design theories. From this perspective, competing alternatives each have distinctive design logic (i.e., responding to different parameters and constraints, focusing on different utility functions, and operating with different command variables). The problem then becomes one of design research: searching the available design alternatives for the best components in developing the best design for the solution. We created a design theory nexus at three levels (fundamental, method-to-construct, instantiation) for evaluating, selecting, and com-
bining the competing alternative approaches to the problem. The design theory nexus provides a viable conceptualization that enables the construction of effective problem-solving artifacts.

Areas for future research develop from our study. We only applied the instantiations of the design theory nexus within IT organizations (or the IT part of organizations). Generally, only IT people were involved. While broader usefulness is promising, the usefulness of the approach in non-IT areas and with non-IT people remains unproven. The IT artifact produced was implemented as a spreadsheet application specifically for each instantiation of the design theory nexus. Based on these successes, it seems possible to develop a general version of the design theory nexus IT artifact that would support the process of creating a design theory nexus artifact. Creating the technological artifact from scratch for each instance of a nexus would no longer be necessary. Further work is needed to design and develop such an artifact.

There is also a need to exercise the design theory nexus as well as the method-to-construct in more and larger organizations. Our experiences have been with a relatively small number of organizations, each involving a relatively small number of people. It takes a great deal of planning and time to exercise an instance of a design theory nexus. In general, the intractable nature of the ideal problem setting for such a nexus, and the complex nature of the underlying and competing design theories, does not lend itself to abstract experiments or surveys. The involvement of real organizations is necessary, requiring senior management commitment and a substantial amount of work to arrange the research.

There are many implications that can be drawn from the promising results of this study. For example, the academic discipline of IS has been criticized for producing research for which the direct relevance for practice is unclear. To a degree, this research might become more clearly relevant if organized for practical application through the use of a design theory nexus in certain problem areas where complex and conflicting IS theories compete in a solution space. The design theory nexus might be useful for directly associating intractable IS problems with potential solutions found in the IS research literature.

While further research is needed to demonstrate further areas and purposes for the design theory nexus, it is clearly a promising and general solution for the selection of an ideal problem solving approach where the context includes many competing, highly dissimilar alternatives. The nexus connects theory to theory, and theory to practice, with highly pragmatic results. We believe it to be an interesting and valuable exemplar of design science research.

Acknowledgments

We thank the ATP, Danske Bank, PBS, SimCorp, and DELTA Software Engineering for allowing us to develop and evaluate the design theory nexus for organizational change. We gratefully acknowledge the research assistance of Pia Eder and Lise L. Hansen for their assistance in developing and evaluating the user involvement nexus described in this paper.

References


About the Authors

Jan Pries-Heje is Professor of Information Systems and head of the User-Driven IT innovation research group in the Department of Communication, Business and IT, Roskilde University, Denmark. His research specializes in organizational and managerial aspects of IT development, such as software process improvement, software development, and—most recently—design and design research. He has published in these areas in journals including Annals of Software Engineering, Information Systems Journal, MIS Quarterly, and European Journal of Information Systems. He is the Danish national representative to the International Federation for Information Processing’s Technical Committee 8 (TC 8) on Information Processing's Technical Committee 8 (TC 8).
Richard Baskerville is Board of Advisors Professor in Computer Information Systems at Georgia State University. His research specializes in security of information systems, methods of information systems design and development, and the interaction of information systems and organizations. His interests in methods extend to qualitative research methods. Baskerville is the author of Designing Information Systems Security (New York: John Wiley) and more than 100 articles in scholarly journals, practitioner magazines, and edited books. He is editor-in-chief of European Journal of Information Systems, and associated with the editorial boards of Information Systems Journal and Journal of Database Management. Baskerville’s practical and consulting experience includes advanced information system designs for the U.S. Defense and Energy departments. He is a former chair of the department at Georgia State and of the IFIP Working Group WG 8.2.

Appendix A

The Spreadsheet Form Used to Measure Actual Conditions in a Project for the User Involvement Nexus

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Answer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Developers have no knowledge of user’s tasks</td>
<td>2 = agree</td>
<td>0 = disagree</td>
</tr>
<tr>
<td>System needs to solve many diverse tasks</td>
<td>2 = agree</td>
<td>0 = disagree</td>
</tr>
<tr>
<td>More than 36 man months estimated for development</td>
<td>2 = agree; more than 24 months; 1 = partly agree; between 12 &amp; 24 months; 0 = disagree; less than 12 months</td>
<td></td>
</tr>
<tr>
<td>More than 24 calendar months estimated for development</td>
<td>2 = agree; more than 24 months; 1 = partly agree; between 12 &amp; 24 months; 0 = disagree; less than 12 months</td>
<td></td>
</tr>
<tr>
<td>Developers do not know the technology that goes into the future system</td>
<td>2 = agree</td>
<td>0 = disagree</td>
</tr>
<tr>
<td>Many stakeholders are involved in the project</td>
<td>2 = agree</td>
<td>0 = disagree</td>
</tr>
<tr>
<td>The project has many interfaces, for example, to other systems</td>
<td>2 = agree</td>
<td>0 = disagree</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resources</th>
<th>Answer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Project has considerable management support in talk and action as well as in demand for results</td>
<td>2 = agree</td>
<td>0 = disagree</td>
</tr>
<tr>
<td>More than 10% of project resources—monies and persons—are dedicated to user involvement</td>
<td>2 = agree; more than 10%; 1 = partly agree; between 3% and 10%; 0 = disagree</td>
<td></td>
</tr>
<tr>
<td>The project has a fluid deadline that easily can be changed</td>
<td>2 = agree</td>
<td>0 = disagree</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Users</th>
<th>Answer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The system is developed for users for which we either know the names or can get the names</td>
<td>2 = yes; can be named</td>
<td>0 = no; nameless</td>
</tr>
</tbody>
</table>