Design Logic and the Ambiguity Operator

Richard Baskerville[1] and Jan Pries-Heje[2]

[1] Georgia State University
35 Broad Street NW, Atlanta, Ga 30302, USA
Tel.: +1 404.413.7362; Fax: +1 404.413.7394
baskerville@acm.org
[2] Roskilde University
Universitetsvej 1, 4000 Roskilde, Denmark
Tel.: +45 4674 3051
janph@ruc.dk

Abstract. Technological rules are one form of expressing management design activities like organizational design, decision design, and information systems design. However, the notion of a "rule" can imply an unintended over-specification of premises and outcomes. We propose a design logic using the concept of an *ambiguity operator* in the predicate logic format. To test the validity of the ambiguity operator, we used it to express the theory under test in a field experiment. The field experiment demonstrated that the ambiguity operator is both useful and valid in logically capturing the field reality when applying designs expressed in the form of technological rules.

Keywords: Design science, design theory, design logic, technological rules, ambiguity, field experiment.

1 Introduction

The science of design differs from design implementations in that it involves development of general solutions to general problems. This scientific perspective searches for a class of design solutions that is developed to treat a class of design problems [1, 2]. This design science perspective differs from ordinary design in which designers search for a specific design solution to a specific design problem.

Much of the current work in information systems design science research is grounded more-or-less directly on Simon's concept of the science of design [3]. Simon's notion for the operationalization of scientific design involved the use of imperative logic, or at least, a process substitute for imperative logic involving a search through a declarative logic solution space. Despite Simon's foundation in design logic, much of the work that followed him, including much of the information systems design science research, used different foundations for operationalizing design science, and there has been very little treatment of design logic.

This tendency in information systems to bypass Simon's logical operationalization of design science should not be surprising. The information systems discipline is quite naturally focused on systems design. In this arena, generalized design is historically

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 180–193, 2010.

operationalized as methodology rather than design logic. In these methodologies, general designs are often represented as notation, diagram protocols or relational algebra. Design science research is sometimes invoked for its focus on the IT artefact [4] and is regarded as offering a conducive philosophy for a greater practice-orientation in research [5].

In concert with interests in design methodology, information systems research has a high degree of respect for research methodologies. The tendency of the information systems field to focus on “design theory” rather than “design logic” is also unsurprising. Ours is a relatively young field, and the centrality of theory helps legitimate design-oriented research as a scientific pursuit. The prominence of hypotheses and support for deductive models of design evaluation is consistent with this legitimation role for design theory [2, 6].

In contrast to information systems, other areas of management research have employed design science as a vehicle focused on decision design. From this perspective, design science opens avenues to discover general decision programming and heuristics [7] or to crack open intractable design problems like multi-criteria decision making [8]. The rule orientation of this work is more closely linked to design logic, but like information systems, has not specifically adopted expressions in design logic. Instead, the decision designs are focussed on more traditional expressions of decision heuristics, such as rules.

These areas of management research, information systems and decision science, are developing advanced understanding of how our fields relate to the science of design. However, design logic is underdeveloped across the disciplines. This may be because logical expressions of general designs appear too rigid and imperative for application to diverse problems. Unbending design logic is easily discarded as too strict for the "real world". Given that design logic was Simon’s original, central approach to expressing design, this paper explores how expressions of design logic can be made expressly flexible and consequently used to describe generalized design solutions for generalized design problems.

In particular, we introduce the necessity for ambiguity in generalized design solutions as a means for achieving generalizable designs and generalized problem settings. This ambiguity is necessary because each problem setting in organizations is unique to a certain extent. This means that each design solution must also be unique to a certain extent. The ambiguity is necessary in positing a general version of a unique setting or design solution without expressing exactly how this setting or solution might be similar to some other, future, setting or solution.

2 Design Logic in Analytical and Generative Settings

The notion of a design logic is not well explored in the literature. Most work in this area regards electronic circuit logic design, or uses the term informally as a reference to the logic behind designs for communication such as rhetorical design logic [9] or community design [10]. For our purposes we will use the term design logic to refer to a set of formal principles of reasoning employed by designers for creating a design. Because such a set of principles maybe used across a general doss of designs, it falls into the realm of meta design and design science.

At least two general intellectual approaches to design creation have been notable. Analytical design is characterized by its basis on rules. With analytical design, the outcome is determinate, defined by a form of propositional understanding [11]. This is perhaps the ideal for relating science and design. The alternative general intellectual approach is generative design. With generative design, the out come is indeterminate, defined more by the subjective feelings of the designer. Both generative and analytical productions are manifestations of working reason leading to figural schemas, but in quite different ways. But generative productions are those in which the faculties of reason align differently than with the analytical productions that are prized in science [cf. 12]. Generative designs have been associated with Kant's conceptualization of aesthetics: intellectual production, in which faculties of reason and are aligned in fundamentally different relationships that emerge from a momentum of ideas into a figural schema more complete than nature [11]. Over simplifying a bit, analytical designs are calculated, generative designs are invented creatively.

In his conceptualization of a science of design, Simon [3] operated almost exclusively in the realm of analytical design. For Simon, the science of design required an imperative logic, one involving not statements of the way things 'are', but statements of the way things 'should become'. However, Simon dismissed imperative logics as flawed. This may be because systems of imperative logic have a different purpose. Formal imperative logic is more associated with ethics than design [13]. As a substitute he introduced declarative logic within a search process. Essentially, the designer sets up a declarative framework for expressing the design solution, and then searches through various values for the framework elements until a satisfactory design is discovered. Consequently, the search for alternatives is a prominent aspect of design research. This search represents the systematic process for discovering design solutions, or the components of design solutions. This search aspect emphasizes the means-ends operations in design research, and the rationality of allocating resources both to the design process and the representation and acquisition of the constituent elements of the artifact being designed. "Problem-solving systems and design procedures in the real world do not merely assemble problem solutions from components but must search for appropriate assemblies" [3, p. 124].

3 Technological Rules; Analytical versus Heuristic Rules

Joan van Aken developed the notion of technological rules as a direction to take design science for applying to management. He was concerned over the usefulness of the corpus of research into management phenomena. In his view, "academic management research has a serious utilization problem" [7, p. 219]. This viewpoint criticizes management research that is too often descriptive and historical. There is little direct usefulness of such reflective studies for managers facing newer and more current problems. Management research would advance if it becomes less descriptive and more prescriptive and less historical and more design-oriented. Van Aken's [14] work argues that a new form of theory, a design theory consisting of "field-tested and grounded technological rules" [7] offers a design science research approach to management.

Van Aken's notion of design science for management recognizes two possible outputs: artefacts or interventions. Three designs can inhabit in a professional episode. The object-design defines the artefact or intervention. The realization-design is the plan for implementing the artefact or intervention. The process-design is plan for the design process itself. In this sense designing is similar to developing prescriptive knowledge.

Van Aken suggests expressing a design in the form of technological rules: "A technological rule follows the logic of 'if you want to achieve **Y** in situation **Z**, then perform action **X**'. The central element of the rule is action **X**, a general solution concept for a type of field problem" [15, p. 23]. A formal expression of this rule would be,

$$(Z,Y) \rightarrow X$$

In this case, **X** is the imperative "Do **X**". Imperative logic seeks to control human actors. An example of this logic would be, "If you want to achieve user acceptance of a new technology in a situation of user alienation, then adopt a participative design approach". This technological rule might be stated,

((**USER ALIENATION**), (**USER ACCEPTANCE OF A NEW TECHNOLOGY**))

→ **ADOPT A PARTICIPATIVE DESIGN APPROACH**

Van Aken [15] warns that technological rules need grounding, Grounding prevents technological rules from degenerating to a form of instrumentalism which operates with 'just' rules of thumb. "In engineering and in medicine, grounding of technological rules can be done with the laws of nature and other insights from the natural and the life sciences (as well as from insights developed by these design sciences themselves). In management, grounding can be done with insights from the social sciences" (p. 25). No matter how helpful technological rules may be to managers, they are not design science unless they are grounded in a way acceptable to social science.

In the example above, the underlying theory is fundamental socio-technical theory, which establishes that participative approaches build acceptance through involvement and commitment among users because participation in the design decisions empowers users and allows them shared control over their futures.

Like Simon, van Aken is using technological rules in a declarative logical mode that is invoking analytical productions at design time. However, van Aken regards these as algorithmic and deterministic prescriptions and continues the development of technological rules beyond such strict analytical productions. He permits the designer important latitude to interpret both the situation and the action.

"However, many prescriptions in a design science are of a heuristic nature. They can rather be described as 'if you want to achieve Y in situation Z, then something like action X will help'. 'Something like action X', means that the prescription is to be used as a *design exemplar*. A design exemplar is a general prescription which has to be translated to the specific problem at hand; in solving that problem, one has to design a specific variant of that design exemplar." [7, p. 227].

This distinction between algorithmic and heuristic prescriptions, and the notion of the design exemplar opens the designer process for both analytic and generative productions. Generative productions are required to make the evaluations of "something like” because creative invention is required to adapt the prescriptive action X to the exact context at hand.

Operating with completely unambiguous rules seems problematic. Some form of generative function is necessary to permit management designers to adapt the rules to situations. In discussing technological rules Pawson and Tilley [16] raised the issue of generative causality. Generative causality recognizes that the outcome is not caused naturally by the interventions of managers. Rather, the outcome is an intended outcome being sought in such interventions. This is a generative form of the causal relationship.

This perspective focuses on the possible ambiguity in the intervention (X), the action being taken with the aim of developing the desired results. Which of the generative mechanism(s) (the various X alternatives) that are used in an intervention actually produces the outcome in a given context? This question leads to the formulation of the CIMO-logic that can be formulated in the following way, "In this class of problematic Contexts, use this Intervention type to invoke these generative Mechanism(s), to deliver these Outcome(s).” [17, p. 395].

Besides detailing the formulation of technological rules by virtue of the CIMO-logic, Denyer et al. [17] suggest the term ‘design proposition’ instead of ‘technological rule’ arguing that ”the latter term suggests—contrary to our intentions—a rather mechanistic, precise instruction”.

The empirical cases we are reporting below used the technological rules rather than the CIMO-logic. While perhaps less logically comprehensive the technological rules were simpler and more accessible for our cases.

4 Ambiguity Operators – Notions of Design Logic

In order to represent this notion of “something like” action X, an ambiguity operator (~) is introduced into the logical representation of the technological rule.

$$(Z,Y) \rightarrow \sim X$$

Which now represents “if you want to achieve Y in situation Z, then something like action X will help.” This rule represents a design production that contains both analytical and generative elements. The analytical element arises from the core rule,

$$(Z,Y) \rightarrow X$$

The generative element arises in the open ambiguity around the action to be taken. The rule expects the designer to invent an adaptation of action X that depends on the situation. This variated action, ~X, makes the design at least partly generative, requiring a different form of reasoning for deciding how a special form of action X should emerge.

Because human organizations are so multivariate, it may he the case that action ~X is more the norm than action X. For many technological rules the ideal setting for action X may arise rarely, making action~X a necessity for most cases. An example of this logic would be, “If you want to achieve user acceptance of a new technology in a situation of user alienation, then adopt something like a participative design approach”. This technological rule might be stated,

((**USER ALIENATION**), (**USER ACCEPTANCE OF A NEW TECHNOLOGY**))

→ **ADOPT SOMETHING LIKE A PARTICIPATIVE DESIGN APPROACH**

Following from the notion of multivariate human sittings it is also likely that situation Z is also an idealization that will rarely be found in an exact form. Here again the ambiguity operator can be used to represent an adaptive technological rule. "If you want to achieve Y in a situation something like Z, then something like action X will help."

$$(\sim Z, Y) \rightarrow \sim X$$

In the absence of specific deductive logic to help disambiguate **~Z ,** the decision as to whether the situation at hand is in fact something like Z will itself be a generative production requiring the designer to imagine the relevant ways in which the two situations are similar.

Indeed, the technological rule can be made fully ambiguated. If you want to achieve something like Y in a situation something like Z, then something like action X will help."

$$(\sim Z, \sim Y) \rightarrow \sim X$$

Without specific deductive logic to help disambiguate **~Y ,** the decision as to whether the goals at hand are in fact something like Y will demand a generative production requiring the designer to imagine the relevant ways in which the two sets of goals are similar.

Essentially, the ambiguity operators offer the necessary latitude to design logic to be flexible and to permit generative productions that adapt the design logic to the setting.

In addition to permitting generative productions in a design science, ambiguity operators introduce generality into design logic. For example, the core technological rule, has often arisen as an empirical point solution:

$$(Z, Y) \rightarrow X$$

A rule such as , "If you want to achieve user acceptance of a new technology in a situation of user alienation, then adopt a participative design approach" will have arisen as a field experience in which participative design was tried as a way to overcome user alienation. At that time, the ideas embodied in the rule were a solution to a quite pointed, specific, practical problem. This *point solution* has since been advanced as the general rule above. We substitute the design logic for the point logic used in the setting. However, there are differences. In the point logic, the participative design was quite specific, for example, the point logic might have involved assigning users to design teams, user specification review sessions, or prototyping with user experimentation. In the design rule, these point solutions are expressed generally as "participative design".

The ambiguity operator opens up further generality, suggesting that "something like" participative design should operate successfully. This enables the designer to consider alternatives to participative design that may work better in the setting-at-hand, and effectively making participative design an element of some unstated general class of solutions that will need to be conceptualized in the future.

From this perspective the ambiguity operator defers the invention of a new general form to future design scientists.

5 Applying the Ambiguity Operator: A Working Theory

We applied the ambiguity operator initially to express our working theory about our expectations for common patterns of ambiguity in organizational settings. This working theory involved two working propositions:

Working proposition 1: If the situation (Z) is disambiguous, designers are driven to more analytical mental productions, and this will disambiguate the action (X) and the goal (Y).

Working proposition 2: If the situation (~Z) is ambiguous, designers are driven to more generative mental productions and these will tend to ambiguate either or both the action (~X) and the goal (~Y).

If these working propositions hold, common patterns of ambiguity should cluster around four of the eight possible rule patterns:

(Z,Y) → X

(~Z,~Y) → ~X

(~Z,~Y) → X

(~Z,Y) → ~X

If the working propositions hold, the other patterns should be rare:

(Z,~Y) → ~X

(Z,~Y) → X

(Z,Y) → ~X

(~Z,Y) → X

6 Research Method

We examined the validity of the ambiguity operator using a qualitative field experiment to test the working theory. Our central purpose in this experiment was to test the operability of the ambiguity logic to express and test theoretical propositions. *Whether the outcome of the experiment confirms or disconfirms the propositions is less important than the validity and clarity of the logic used to express the propositions and the results (for our purposes).* In the case at hand, as the reader will see, the experiment disconfirms its propositions. The logical clarity with which this result finds expression, and the ability to reformulate the propositions for further result, may indeed provide better evidence for the strength of the ambiguity operation than an alternative result that simply confirmed the theory under test.

In this experiment, practicing information systems project managers explored the ambiguity in their technological rule settings. The purpose of this experiment was to test whether the use of an ambiguity operator as a logical modifier for technological rules would lead to insights into the use of technological rules in real project settings.

6.1 Field Experiment

The technological rules for this setting were derived from the design science nexus [8]. This framework for organizational change was presented to a number of experienced project managers undertaking an executive master in project management and process improvement. The technological rules that follows the 'basic' **(Z,Y) → X** form can be found in [18]. An example is shown in Figure 1.

If you want to initiate organizational change **in a situation** where you:
Believe that formal structures needs change
Where change is needed fast
Then choose a *Commanding* approach where change is driven and dictated by (top) management; one where management takes on the roles as owner, sponsor and change agents.
If you want to initiate organizational change **in a situation** where:
You believe that target group is very diverse and has large individual differences
The target group are experts
Then choose an *Optionality* approach where change is driven by the motivation and need of the individual; it is to a large degree optionality.

Fig. 1. Example technological rules from the organizational change nexus

The organizational change design nexus includes ten different approaches for change each having a separate set of technological rules. The project managers were presented the 10 approaches in a 4-day seminar (40 teaching hours) in November 2009. After the 4-day seminar they were asked to hand in their organizational change plan after having applied the technological rules in their own organization to their own change project. In Table 1 we have shown the participating change projects.

The participants' projects plans were then coded for ambiguity in one or more of the three elements in the technological rules. In the coding we looked for any and every possible sign of ambiguity in Y (aim), Z (situation) and X (action).

6.2 Results from Field Experiment

Table 2 depicts the results of this experiment. We have included quotes from the participants' project plans that clarify how these situations are, or are not, ambiguous.

Table 1. Overview over the 15 project managers, companies and change projects engaged

Name	Company	Change project
1. Alfa	Engineering company that runs projects for customers	Implementing a new project calculation tool for all projects in company
2. Bravo	School which have a drug problem	Change attitude towards drugs; it is not socially acceptable
3. Charlie	Large governmental organization	Implementing a new budget and accounting model
4. Delta	Fusion between two Engineering companies	Integrate the two former companies better
5. Echo	Hospital in the Greater Copenhagen Area	Improve
6. Foxtrot	Software House selling solutions to customers for electronic case and document handling	Implementing solutions for electronic case and document handling at customer sites
7. Golf	Large software house in financial sector	Fusion two companies successfully
8. Hotel	Local Healthcare Center in Copenhagen	Relatively newly established public organization. Needs to start working according to defined goals for local healthcare
9. India	Silviculture Association	Implementing time and material registration
10. Juliet	Intelligence Service	Teach personal to be able to cope with modern terrorism
11. Kilo	Large Danish Municipality	Changing work processes and work routines in part of the organization
12. Lima	Smaller hospital close to Copenhagen	Changing the ways patients are treated at the hospital
13. Mike	Danish Parliament	Changing workflows
14. Novemb	Psychiatric and handicap administration in a Region of Denmark	Optimizing workflows
15. Oscar	Major Bank	Change projects in general

Table 2. Coding of ambiguity for the 15 cases in the field experiment

	You want to achieve Y	in situation Z,	then perform action X
1. Alfa	Y very clear; Implement a tool	Z clear. Consistent number of customers	**X ambiguous;** “Specialist driven change … I cannot see that” “Combined with the optionality change strategy it is more meaningful” “Model needs to be seen either in relation to internal or external customers”

Table 2. (*Continued*)

2. Bravo	**Y ambiguous;** "… present *a number of ways* that the individual can identify with"	Z clear; Students with drug problems	X relatively clear; Combine optionality with a specialist driven strategy "… to ensure that the individually preferred methods are analyzed, diffused and in fact used."
3. Charlie	Y somewhat clear; Implement new budget model before the end of 2010	**Z ambiguous;** "Large geographical distribution" "… it is not sufficiently taken into account how large a part of the organization that is involved"	X clear, Commanding will fit well with hierarchical structure. Combine with BPR-strategy to ensure that the old budget model is extinguished
4. Delta	Y clear; Better integration to achieve the benefits of being together	Z clear; Two companies brought together as one (fusion) a year ago	X seems to work; "Socializing and specialist driven "seems to work fine here a year after the fusion …" "Commanding scores low. That was the strategy used the first period after the fusion. It is clear that it did not work"
5. Echo	**Y ambiguous;** "It *may* the tasked to management to help staff create *a room* for ideas and creativity"	**Z ambiguous;** "The hospital has large cultural *differences* from one department to another"	**X ambiguous;** "The tool [change nexus] is *somewhat useful*. I will not suggest change strategies based on this tool alone"
6. Foxtrot	Y very clear; Implement the tool sold	**Z ambiguous;** "Major difference between a strategic change for a whole organization, and a change in one or few departments"	**X ambiguous;** "I believe the type of change has as much to say as the situation" "You would probably score differently if you had has mapped the situation right before using the Nexus"

Table 2. (*Continued*)

7. Golf	**Y ambiguous;** "The new organization faces a relative long process where they choose infrastructure and work processes for future systems development"	Z clear; Company has just agreed to fusion with another	**X ambiguous;** "It seems to be a very flimsy foundation to base a change strategy on 2 to 4 parameters"
8. Hotel	Y clear, Start working as planned	Z clear; Newly established. Goals for local health-care defined from central healthcare	**X somewhat ambiguous;** "Model is useful but before the final decision on strategy one needs to think in more parameters and not be seduced by the one-string structure [of the Nexus]"
9. India	**Y ambiguous;** "It is our wish to undertake adjustments that can increase profit …"	**Z ambiguous;** "It is a fact the organization has responded very slowly to using time and material registration"	X clear; [BPR strategy scoring highest]. "I am convinced that this model is useful"
10. Juliet	Y clear; To be able to cope with modern terrorists	Z clear; The "modus operandi" of modern terrorists is known	X became clear; "Originally I was skeptical... but the model ashamed me …" "The model recommended the strategy that I would intuitively have chosen"
11. Kilo	**Y ambiguous;** "… it is the insights of the specialists that will be used to optimize workflows and higher quality"	**Z ambiguous;** "There are large differences between tasks in the organization so it is natural that it will take time to implement changes"	**X but with some ambiguity;** "Tool is easy to use and a good help … will work fine in combination with other models so you don't base your strategy choice on just one model"
12. Lima	Y somewhat clear; "We do what we have always done" "The environment is characterized by carelessness"	**Z ambiguous;** "I believe the model doesn't paint a true and fair view of the organization"	X somewhat clear; "The nexus creates some overview and some recommendations that fits well"

Table 2. (*Continued*)

13. Mike	Y clear; New workflows	**Z ambiguous;** "the divergence between the units means that the change doesn't have to look the same in different places"	X clear; "I had my boss fill out the Nexus questionnaire, and he came up with the same"
14. Novemb	Y clear, A template for administrative consideration of cases	Z clear; Long and tedious case workflows today	**X ambiguous;** Argues that one cannot uncritically use the results; and argues for another than the model
15. Oscar	**Y ambiguous;** "… without having specific changes in mind."	**Z ambiguous;** Many knowledge workers and "highly diverse departments"	X clear; [Optionality strategy scoring highest]. "… fits well with the organization."

From these data in table 2, we can see that there were four rule patterns that occurred multiple times

(Z,Y) → ~X (appears three times)

(~Z,Y) → X (appears three times)

(~Z,~Y) → X (appears twice)

(Z,Y) → X (appears twice)

The remaining rule patterns that occurred once each:

(~Z,~Y) → ~X

(~Z,Y) → ~X

(Z,~Y) → ~X

(Z,~Y) → X

7 Discussion

The field experiment offers results at two levels. The first level regards the outcome in terms of the propositions that were disconfirmed by the data. While this is not our primary objective, we would be amiss not to discuss these results in terms of their implications for future research. This discussion will lead to the results from second level. The second level regards the validity and usefulness of the ambiguity operator when used to modify the predicate logic of technological design rules.

7.1 Discussion of the Field Experiment

This data does NOT support the working propositions and consequently do NOT support the working theory. Indeed, the most common patterns appeared from those expected to be rare. Instead, it appears that the data align better with an extension of the generative causality and CIMO logic discussed earlier. Recall that the CIMO logic emphasized the generative causality arising from the ambiguity in actions that may be taken in an intervention. This situation is a common pattern in our data, similar to one in which the action to be taken is ambiguous, (Z,Y) → ~X.

However, the data suggest that this logic may need to be expanded. The other most-common pattern involved ambiguity in the situation (Z). We can speculate that ambiguity in Z drives analytical treatment of Y and X, (~Z,Y) → X.

7.2 Discussion of the Ambiguity Operator

While this simple qualitative field experiment is insufficient to prove or disprove these theories, it does strongly validate the usefulness of the ambiguity operator in design logic. As we mentioned earlier, whether the outcome of the experiment confirms or disconfirms the propositions is less important than the validity and clarity of the logic used to express the propositions and the results.

The ambiguity operator permits a clearer understanding and representation of the expected and unexpected variation between the elements of a design logic expression and an instance of the application of this design logic. In this experiment, the ambiguity operator enabled us to clearly express a working theory and its system of propositions. Based on these expressions we were able to formulate and execute a simple, qualitative field experiment. Surprisingly, the results of the experiment upsets this theory, an outcome that perhaps provides more convincing evidence of the value of the ambiguity operator than a confirmation of the theory. We find this evidence especially convincing because of the immediate ability to use the logic to reformulate new propositions for further research.

8 Conclusion

Design is a core part of many management activities be it organizational design, decision design, information systems design, etc. Technological *rules* can be used to express a design. But the word rule is often mis-interpreted as something that should be strictly followed.

In this paper we have shown that an ambiguity operator can be added to the basic formulation of technological rules thereby formulating a design logic in predicate logic format. Indeed, technological rules can be made fully ambiguated in the following way: If you want to achieve something like Y in a situation something like Z, then something like action X will help."

(~Z,~Y) → ~X

We examined the validity of the ambiguity operator using a field experiment with 15 people. We expected that if the situation (Z) is disambiguous, designers are driven to more analytical mental productions, and if the situation (~Z) is ambiguous, designers are driven to more generative mental productions and these will tend to ambiguate either or both the action (~X) and the goal (~Y). None of these working propositions were supported.

Instead we found that the data align better with an extension of generative causality and the so-called CIMO-logic expanded.

Thus our conclusion is that the concept of the an ambiguity operator seems to be very useful in representing what really happens when a design expressed in the form of design logic meets reality in the field. Because it makes design logic more diversely applicable, the ambiguity operator makes design logic a more useful tool for the purpose of design science research.

References

1. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
2. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an information system design theory for vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
3. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
4. Orlikowski, W.J., Iacono, C.S.: Research commentary: Desperately seeking "IT" in IT research - A call to theorizing the IT artifact. Information Systems Research 12(2), 121–134 (2001)
5. Iivari, J.: A paradigmatic analysis of Information Systems as a design science. Scandinavian Journal of Information Systems 19(2), 39–63 (2007)
6. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)
7. van Aken, J.E.: Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. The Journal of Management Studies 41(2), 219–246 (2004)
8. Pries-Heje, J., Baskerville, R.: The design theory nexus. MIS Quarterly 32(4), 731–755 (2008)
9. Bryan, S.P.: Cognitive complexity, transformational leadership, and organizational outcomes, Louisiana State University and Agricultural & Mechanical College: United States – Louisiana (2002)
10. Fritzen, S.A.: Can the Design of Community-Driven Development Reduce the Risk of Elite Capture? Evidence from Indonesia. World Development 35(8), 1359–1375 (2007)
11. Groat, L., Wang, D.: Architectural Research Methods. Wiley, New York (2002)
12. Kant, I.: The Critique of Pure Reason (1781). In: Rand, B. (ed.) Modern Classical Philosophers, pp. 370–456. Houghton Mifflin, Cambridge (1908)
13. Gensler, H.J.: Formal Ethics. Routledge, New York (1996)
14. van Aken, J.E.: Valid knowledge for the professional design of large and complex design processes. Design Studies 26(4), 379–404 (2005)
15. van Aken, J.E.: Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. Journal of Management 16(1), 19–36 (2005)
16. Pawson, R., Tilley, N.: Realistic Evaluation. Sage, London (1997)
17. Denyer, D., Tranfield, D., van Aken, J.: Developing Design Propositions through Research Synthesis. Organization Studies 29(3), 393–413 (2008)
18. Pries-Heje, J., Baskerville, R.: Management Design Theories. In: Pries-Heje, J., et al. (eds.) Human Benefits Through the Diffusion of Information Systems Design Science Research. Springer, Boston (2010)