IS Design Pedagogy: A Special Ontology and Prospects for Curricula

Waguespack, Leslie J.
lwaguespack@bentley.edu
Computer Information Systems
Bentley University
Waltham, MA 02452

Abstract

Design as an academic endeavor has a rich history in the visual and performing arts as well as the "construction" arts: architectural, industrial, graphic, interior, fashion, and landscape design. Design in the natural and commercial sciences is largely peripheral. Although not ignored, design in the sciences predominates as a dialect of problem-solving rather than artifact creation. Information Systems (IS) as a fusion of computer science and business struggles with the identity and role of design as it leans heavily on its roots in mathematics and formal logic with the scattered influences of statistics (data analysis) as practiced in research among the social and behavioral sciences. Design as a practical skill is a critical ingredient in successful information systems. Yet, design as a distinctive element in programs and curricula of IS and computing is (at best) haphazardly diffused – if not completely omitted. This paper presents a special ontology of design to frame the opportunities and justification for conscious and deliberate design pedagogy in IS and computing education. It presents an example of integrating design pedagogy into existing object modeling and data management syllabi by tailoring design quality guidelines to the specific paradigm.

Keywords: Information Systems Curriculum, Information Systems Design, Thriving Systems Theory, and Special Ontology of Design.

1. INTRODUCTION

Information systems engage myriad issues: audience, purpose, function, materials, and economics. Successful development requires gathering alternatives to be assessed, weighed, balanced and selected to eventually result in a deliverable – a confluence of design choices that *is* the system – *design-as-a-noun*. It is curious that *design-as-a-verb* is not a core focus in contemporary information systems (IS) education. The closest approximation to design as a learning goal is perhaps *problem solving*. If anything, design as a goal in IS curricula has diminished rather than grown. If we inspect IS model curricula as surrogates for defining the discipline it is clear that "… [the] distinction between design and implementation has faded from the structure of computing education. To ignore the conceptual distinction between the *design* and an *implementation* is tantamount to accepting any 'solution' without even considering [quality]…" (Waguespack, 2011)

Within the business domain that envelopes IS education, the theme of "design" has become *de rigueur*, particularly as it relates to creativity (Cohen, 2014). Design expertise is recognized in industry as a recognized competitive advantage. Apple Inc. is a revered corporate leader, not just in technology, nor just as an innovator in the marketplace, but particularly because of its marked, tenacious, and steadfast focus on the importance of design (Turner, 2007). Design thinking is a prominent tool in solving social as well as commercial challenges. "… [L]arge organizations such as the Bill and Melinda Gates Foundation, the Rockefeller Foundation, the Hewlett Foundation, and others, have enthusiastically embraced design thinking. At the same time, non-profit design companies like D-Rev, Design that Matters, … IDEO.org, and others are collaborating with social entrepreneurs and NGOs to bring exciting new innovations to those most in need. For perhaps the first time in the history of design, it's possible to make a career designing for the social sector" (Brown, 2009).

Design has long been held in esteem in the arts, and particularly in architecture (Alexander, 2002). Design taps into the human capacity for emotive response (aesthetics); not readily evoked by mathematics or algorithm. All but ignored in computing education, design's value is primarily relegated to achieving computing artifacts that "work." But, in this current, technologically based culture, the distinction between a system that "works" and one that "works well!" determines the success or failure of websites, mobile applications, and enterprise systems. The collection of design choices that form a computing artifact produces a stakeholder experience that may range wildly from unacceptable to elegant depending on the skill and insights of the designer(s). There may have been an era when "functional" was an adequate and acceptable level of design quality. But that era has past. Today's individual, corporate, or governmental consumer expects computing artifacts that "delight" by delivering a level of satisfaction that not only "works" but, anticipates, simplifies, and empowers the users at their task. IS professionals must be able to partner in the organizational role of design; thus design, both as noun *and* as verb, must be integral to computing education.

This paper presents a special ontology of IS design as a pedagogical platform for understanding and integrating design in IS and computing curricula. This paper proceeds as follows: First, there is a brief, selected review of design research in computing. Next perspectives on design in the sciences, humanities and design thinking are contrasted. A special ontology of design defines the objects and actions behind a descriptive narrative of design behavior. There is an example of integrating design pedagogy in existing IS courses using Thriving Systems Theory in a metaphorical lens on design quality. Finally there is a brief reflection of the ongoing pedagogy experiment using these ideas and future directions.

2. DESIGN RESEARCH IN IS

In IS research there is a renewed interest in design; a recognition that design quality should not be insignificant or accidental in systems development. Design Science research has become a movement (Hevner & Chatterjee, 2010) and Information Systems Design Theory (ISDT) promises to formalize quality systems assessment (Walls, 2004, Gregor, 2007).

Design in the object-oriented paradigm evolved from art and physical architecture in Christopher Alexander's pattern languages and the notion of design patterns (Alexander, 1977, 1979; Gamma et al., 1995). Alexander identified the "Quality without a Name," or perhaps, a *je ne sais quoi* capturing the essence of designing – "to speak of design is to speak of quality" (Alexander, 1979). Alexander's theory of *living structure* inspired Thriving Systems Theory of design quality in information systems (Alexander, 2002; Waguespack, 2010; Waguespack & Schiano, 2012, 2013). There is an arc of design influence from Christopher Alexander, to the "Gang of Four," to Ward Cunningham and Kent Beck, as manifested in object-orientation, the Unified Modeling Language, design patterns, and agile methodologies (Beck et al., 2001).

3. DESIGN: ARTIFACT AND EMERGENCE

There are widely varied conceptions of design across three cultures of theory: the natural sciences, the humanities, and design thinking (Simon, 1966). (See Table 1 below adapted from Cross, 2007.)

	Phenomenon	Methods	Values
Science	The natural world	Controlled experiment, classification, analysis	Objectivity, rationality, neutrality, "truth"
Humanities	Human experience	Analogy, metaphor, evaluation	Subjectivity, imagination, commitment, "justice"
Design	The artificial world	Modeling, pattern-formation, synthesis	Practicality, ingenuity, empathy, "appropriateness"

Table 1. Conceptions on Design

The natural sciences dwell on "why" objects in nature exist as they do – basically taking intact, functioning "objects" apart to see what they are made of and how they work. Objects are accepted, as they are, independent of human intention or judgment. "The 'value' of an object in the natural sciences view vests in its existence and/or survival with any human satisfaction based on 'accident of nature' (Babb & Waguespack, 2014).

The humanities, on the other hand, ascribe an artifact's "value" to intention and judgment in the human encounter with the artifact that prompts a psychological or emotional response (i.e. a degree of satisfaction) – specific to the observer. This perspective contemplates the observer's mindset and expectations. But as with the natural science perspective the artifact is perceived as extant – valued for its *design-as-a-noun* with little or no consideration of the artifact's origin or emergence.

The heritage of IS's treatment on design closely aligns with the positivist philosophy of mechanistic or mathematical artifacts, extant phenomena as in natural science – the essence of these phenomena existing independent of human judgments and devoid of aesthetic quality. Design science research has incorporated a human dimension to *design-as-a-noun* but, treats computing artifacts fundamentally as "block boxes," extant and similar to natural phenomena, but evaluable in human terms.

Unlike the natural sciences or humanities, *design thinking* champions the creative aspect of design – *design-as-a-verb*.

> "The central idea of design is 'the conception and realization of new things'. It encompasses the appreciation of 'material culture' and the application of 'the arts of planning, inventing, making and doing. At its core is the 'language' of 'modeling;' it is possible to develop students' aptitudes in this 'language', equivalent to aptitudes in the 'language' of the sciences (numeracy) and 'language' of humanities (literacy). Design has its own distinct 'things to know, ways of knowing them, and ways of finding out about them'. … That is the distinctive character of a *designer*." (Cross, 2007, p.17).

This conception of design resonates with the core competency that identifies IS professionals – *conceiving and crafting systems*. This competency is the focus of the quality design pedagogy addressed in this paper.

> "[Design] quality in IS artifacts entails: 1) a grasp of functional needs, 2) an aesthetic sensibility attuned to the stakeholder(s)' perception of quality and 3) the skill to engage technology that allows a formulation of (1) which allows (2) to resonate. Design in this formulation of quality is central to the entire IS discipline: technology, society, organization, management, and operation – every relevant aspect of IS" (Babb & Waguespack, 2014).

4. A SPECIAL ONTOLOGY OF IS DESIGN

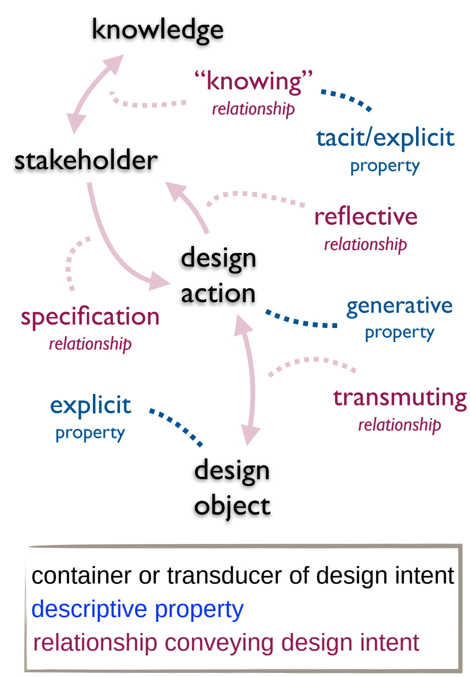

Figure 1 - Special Ontology Of Design

Design is often portrayed as a discrete phase in the systems development life cycle (SDLC). It actually permeates the entire SDLC. Design, as the "conception and realization of new things," is integral to every SDLC work product, both individually and in composition by acts of *design-as-a-verb*. Every work product results from design choices that shape, include, exclude, and juxtapose various alternatives in policy, documentation, specification, interfaces, algorithms, program code, procedures, acceptance and performance standards, protocols, software, training exercises, contracts, etc. Design behavior involves a cycle of decomposing the design space and compositing design choices in an intrinsically recursive activity. A special ontology follows that enumerates the objects and actions characterizing design.

Special ontologies identify individuals, attributes, relationships, and classes defining relevant concepts of interest that establish a framework for reasoning within a specific domain. The special ontology that follows explains a general theory of design. (See Figure 1.)

(Gero, 1990) and (Gero & Kannengiesser, 2002) offer ontologies focused specifically on design element manipulation.

Individuals – *Stakeholders*, *knowledge*, *design objects*, and *design actions* comprise this special ontology of design. A *stakeholder* is a human agent invested in creating an artifact (i.e. client, owner, user, designer, or consumer). *Knowledge* denotes the sum of facts and skills personally accessible to a stakeholder. A *design object* is an explicit proposition of stakeholder intention, a *model*: a) directly specified by the stakeholder, or b) the result of a design action. A *design action* is a generative activity that: a) creates a design object specifying stakeholder intention, b) transmutes a design object from one form into another as a step in fabricating an artifact, or c) delivers a design object to a stakeholder for reflection.

Attributes (properties) – Design objects record *explicit* knowledge (i.e. published). Stakeholders access their knowledge through *explicit* or *tacit* "*knowing*." A stakeholder can specify/explain their *explicit* knowledge (i.e. knowledge acquired through formal education) and be aware of but, not be able to specify/explain their *tacit* knowledge (i.e. knowledge acquired through their personal experience of "living"). This is the distinction between knowing "what" and knowing "how" (i.e. "*We know more than we can tell*") (Polanyi, 1966). Design actions are *generative*.

Relationships - Stakeholders associate with their knowledge through an *explicit* or *tacit* "*knowing*" relationship, and with design actions through *specification* and *reflective* relationships. Design actions associate with design objects as input or output through a *transmuting* relationship.

Classes distinguish generative design actions as *specification, transmuting,* or *reflective*. *Specification* publishes stakeholder intentions by creating a design object. A *transmuting* design action reshapes an existing design object from one form into another in its progression toward the target artifact (e.g. a stakeholder intention is specified as a design object, which is transmuted to become a requirement specification, which is transmuted to become a design specification, which is transmuted to become a prototype, etc., etc.). A *reflective* design action provides a stakeholder access to a design object for reflective evaluation, to assess a degree of satisfactory progress (or possible completion) (Schön, 1983, p. 271).

A normative depiction of design behavior follows based upon this special ontology of design (Figure 1 above).

5. A PROTOTYPICAL DESIGN NARRATIVE

Design behavior is characterized by three intrinsic concerns: "*why*," "*how*," and "*what*." (See Figure 2 below.)

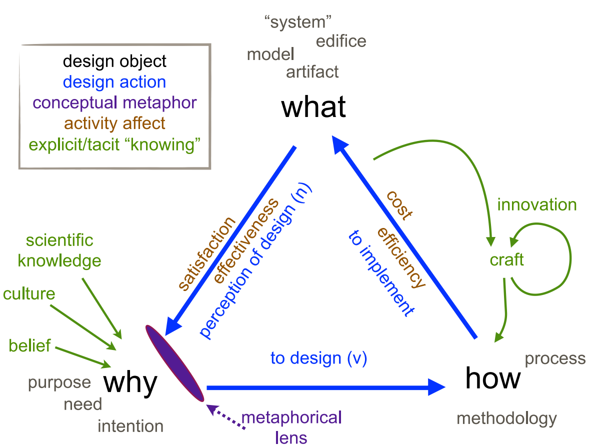

Figure 2 - Design Behavior

"Why" encompasses the intentions that originate from stakeholders and motivate the design process – the "qualities" that describe the desired artifact. These qualities take the form of needs or purpose that reflect a call for change in the status quo – in general creating a "new experience" for the stakeholder(s). The qualities prescribed in the "*why*" are guided, directed, and/or constrained by the stakeholders' mindset, their world-view. That mindset reflects the combined explicit and tacit knowledge sourced from formal education and personal experience. Conscientious stakeholders try to act objectively by eschewing unnecessary or irrelevant influences on their specification of "*why*." But, human beings find it sorely difficult to achieve total objectivity in their world-view shaped by culture, geography, politics and beliefs inculcated through family, society, and life experience. As the stakeholder(s)' intentions enter into the "*why*" of design, they are necessarily colored by a personal world-view depicted in Figure 2 as a *metaphorical lens*.

The **first action of design behavior** is creating a design object, a *requirements* proposition, ostensibly accurately and fully representing the stakeholder(s)' "*why*." A sincerely objective expression of the "*why*" should avoid artifact implementation details. They are poor substitutes for a clear statement of desired artifact qualities.

"How" – The **second design action** (*design-as-a-verb*) transmutes the "*why*" into a plan of instructions, materials, and protocols to be followed/executed using available tools and skills to construct an artifact. Unless those tools and skills reflect heretofore unknowns, implementation is an execution of *craft*. *Craft* is acquired through successful practice on similar implementation projects and assures predictable outcomes with reliable cost and quality. *Craft* strongly influences design choices in transmuting requirements into a plan of construction. *Innovation* seeks to improve *craft* – reduce costs and increase efficiency. With an implementation plan in hand the construction steps are executed to produce a provisional artifact.

"What" is that provisional product of construction aspired to in the *requirements* proposition. The product may be material or informational: document, program, system, or, perhaps, machine or edifice. Regardless, only at this point can its value be assessed. It is provisional because a perfect alignment of artifact qualities with stakeholder intentions is a veritable impossibility. Artifact acceptability is inevitably "satisficing" (to some degree) (Simon, 1996 p. 119).
The **third design action** is reflective, evaluating the "*what*." The stakeholder(s) compares the qualities they perceive in the artifact (*design-as-a-noun)* against their understanding of intention in the "*why*." The degree of perceived equivalence determines *satisfaction*. The same *metaphorical lens* used to shape the "*why*" mediates the stakeholder(s) perception of the "*what*" – this time in reflection rather than specification (Schön, 1983, p. 271). In an iterative design process unacceptable misalignment of the "*what*" with the "*why*" prompts rethinking and/or adjustments to requirements (clarifications, corrections, additions, etc.) with a repeat of the cycle of design actions in order to improve stakeholder(s) *satisfaction*.

6. PROSPECTS FOR CURRICULA

Although IS model curricula have paid virtually no attention specifically to design (Waguespack, 2011), its pervasive impact across every aspect of the SDLC (and the Agile variants vying to supplant the SDLC) manifest the need to acknowledge design's importance and to find its place in IS and computing curricula. By "factoring" the design narrative above potential focus areas emerge.

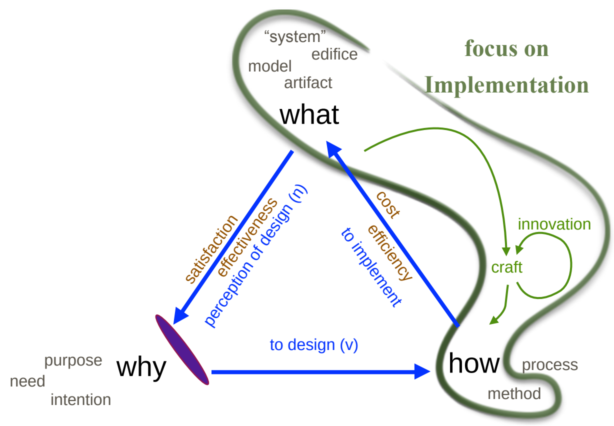

Figure 3 - Implementation Pedagogy Focus

Implementation focus - Programming languages, application platforms, and pattern languages mark a focus on implementation that commonly constitutes the software development dimension of computing pedagogy. (See Figure 3 above.) While the model curricula focus primarily on syntax and coding, particularly computer science and computer engineering, more emphasis is needed on design.

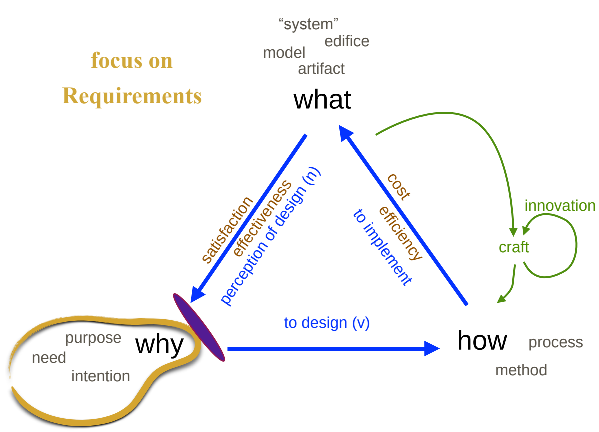

Figure 4 - Requirements Pedagogy Focus

Requirements focus – Requirements engineering is preoccupied with business rules and organizational interdependencies. This focus is often deemphasized in computer science curricula while it predominates IS curricula housed in schools of business. (See Figure 4 above.) The pivotal role of the *metaphorical lens* can be a balance point for analyzing business model and information system alignment.

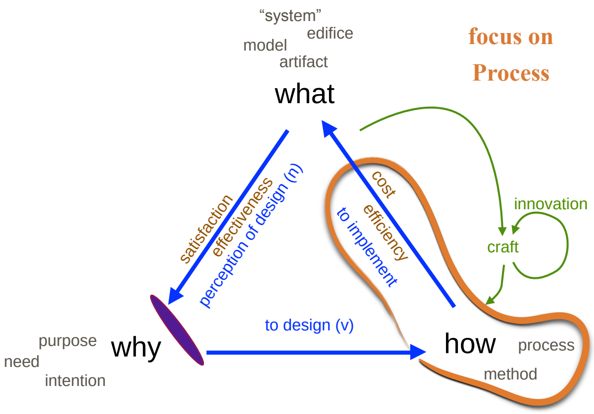

Figure 5 - Process Pedagogy Focus

Process focus – Software engineering (in particular) emphasizes life cycles, methodologies, and metrics. Team and project productivity are clearly impacted by "*why/how*" alignment in both project and process management. (See Figure 5 below.) Concern for aesthetic qualities can improve documentation, reuse, training, customer support, and maintenance – all aspects of cost of ownership.

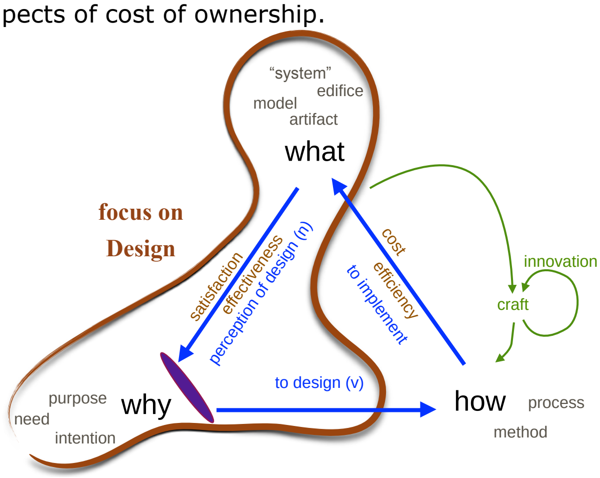

Figure 6 - Design Pedagogy Focus

Design focus – *"Design"* as a discipline encompasses design thinking, creativity, and reflective practice that shape a comprehensive professional design competency (Schön, 1983, p. 271). The *metaphorical lens* is fundamental to designer formation. Design pedagogy should nurture a design mindset through readings, exercises, and projects (individual and team) that develop the student's "design muscle memory" by ingraining a tacit "*knowing*" of design quality through practice and reflection with IS tools and methods guided by master designers. (See Figure 6 below.)

These four "slices" of design focus can only hint at the possibilities. The next section outlines an example of an integration experiment across three course syllabi.

7. DESIGN INTEGRATION STRATEGY

A strategy for design pedagogy must navigate the constraints of graduation requirements, accreditation and the general politics that impact program design in collegiate curricula.

Add on – As is the case with many topic areas perceived as value-added to degree programs in higher education, "add a course" to the graduation requirements appears to be the most straightforward approach. If this is feasible, the opportunity for design coverage may be quite broad. If a school of design is available, a general survey of design principles may be appropriate. Practically speaking, adding a course is more often not feasible.

Integrate – Integrating design pedagogy into existing coursework is the alternative to adding on a new course. For example, for the past five years a design integration one IS program has been experimenting with integrating design pedagogy in three existing course syllabi: Business Systems Analysis and Modeling (UG), Object-Oriented Systems Analysis and Design (GR) and Data Management and System Modeling (GR). These courses involve constructing SDLC artifacts (i.e. requirement specifications, data models, database schema, SQL queries, transactions, UML models, etc.). Each is a three-credit hour one-semester course packed tightly with learning objectives. Students in courses like these naturally fixate on objective issues: syntax, query results, error-free execution, etc. Any aesthetic flavor of design quality is a rather exotic concept to them. So, the goal of integrated design pedagogy in these courses is to instill an aesthetic sense of design quality.

> "In order to formulate a design problem to be solved, the designer must frame a problematic design situation: set its

boundaries, select particular things and relations for attention, and impose on the situation a coherence that guides subsequent moves" (Schön, 1983).

Choice Property	Design Action	Action Definition
Modularization	elaborate	develop or present (a theory, policy, or system) in detail
Cohesion	factor	express as a product of factors
Encapsulation	encapsulate	enclose the essential features of something succinctly by a protective coating or membrane
Composition of Function	extend	render something capable of expansion in scope, effect, or meaning
Stepwise Refinement	modularize	employing or involving a module or modules as the basis of design or construction
Scale	align	put (things) into correct or appropriate relative positions
Identity	expose	reveal the presence of (a quality or feeling)
Patterns	assemble	fit together the separate component parts of (a machine or other object)
Programmability	identify	establish or indicate who or what (someone or something) is
User Friendliness	focus	(of a person or their eyes) adapt to the prevailing level of light [abstraction] and become able to see clearly
Reliability	accommodate	fit in with the wishes or needs of
Correctness	pattern	give a regular or intelligible form to
Transparency	generalize	make or become more widely or generally applicable
Extensibility	normalize	make something more normal, which typically means conforming to some regularity or rule
Elegance	coordinate	bring the different elements of (a complex activity or organization) into a relationship that is efficient or harmonious

Table 2 – TST Choice Properties

The objective is to expand the student's *metaphorical lens* by imprinting aesthetic sensibility as part of that *coherence* to guide their design choices. The design choice properties defined in Thriving Systems Theory (TST) provide the quality framework in this experiment (Waguespack, 2010, Waguespack & Schiano, 2013). TST "seeds" their *metaphorical lens* with design quality concepts for *naming* and *framing* design elements and actions: a) specifying stakeholder intentions, b) preserving stakeholder intentions in design actions, and c) evaluating artifact features in "reflective conversation" (Schön, 1983, p. 271).

Thriving Systems Theory defines fifteen properties underpinning quality choices in computing artifact design. (See Table 2 above.) Each property is strengthened by a generic action that when applied properly enhances design quality. The first six properties differentiate structural design features while the remaining nine differentiate aesthetic aspects that together impart a sense of stakeholder satisfaction – a sense that the artifact serves as intended, as expected, or is "as it should be."

The IS or computing professional will readily recognize the first six properties in Table 2 as desirable properties in computing structures like documentation, source code, abstract data types, data models, or modularity. They are all primarily static characteristics amenable to counting, measuring, i.e. – numeracy! The remaining nine properties basically defy numeracy because they prompt personal, emotional, or psychological observer reactions – reactions mediated by the observer's personal disposition and knowledge, their world-view. Every design choice exhibits each of these fifteen properties with a subtlety that renders it near invisible, a pronounced conspicuousness, or to some degree in between. Although individually present, the properties are more often perceived in confluence, in a broader sense of design quality. Discerning design quality is akin to recognizing a person's face, physiognomy, even when the human observer is unable to distinguish or quantify specific facial features that confirm the subject's identity. This is "tacit knowing," a demonstrable human competency, accomplished without specifiable knowledge, a practical skill acquired through personal experience. (Polanyi, 1966, p.17)

Design quality clusters – TST comprises fifteen choice properties of design inscribed along the circle's circumference in Figure 7 above. The confluence of the six properties of design structure appears in the convergence of the pair-wise combination of their property affects – shaded in pink. The six properties cluster to articulate progressively complex, structural qualities described more thoroughly in (Waguespack, 2010). The remaining nine properties likewise cluster to articulate progressively complex aesthetic qualities more readily interpreted through analogy rather than numeracy – shaded in green. In progressive pair-wise composition the clusters *frame* and *name* the stakeholder's impression of the artifact in their learning of it, their use of it, and adapting/modifying it, – where the stakeholder is *indwelling* with the artifact, assimilating a sense of the artifact as part of the stakeholder's "world."

> "The use of the term 'indwelling' applies here in a logical sense as affirming that the parts of the external world that we [assimilate] function in the same way as our body functions when [skill becomes second nature]. In this sense we live also in the tools and probes which we use, and likewise in our intellectual tools and probes." Mitchell quotes Polanyi (Mitchell, 2006).

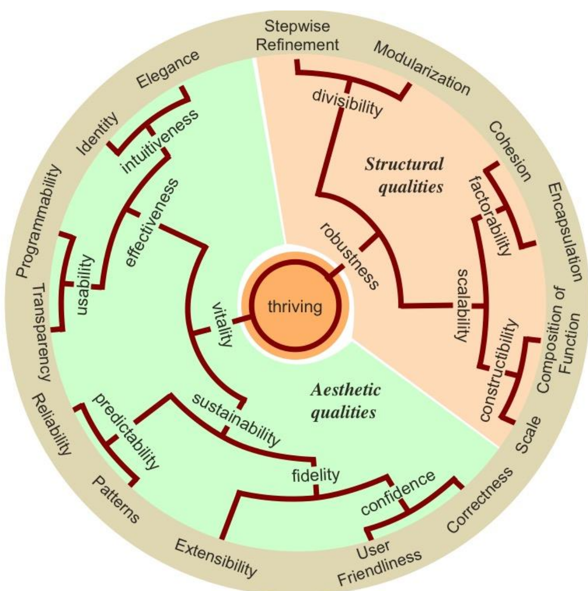

Figure 7. Design Quality Clusters

Accommodating multiple paradigms – The design actions strengthening choice properties in Table 2 are generic – paradigm independent. The structural and aesthetic qualities depicted in Figure 7 are explicated by those generic design actions. But, students must learn to build artifacts using tools and objects specific to a particular modeling paradigm with its own design space "language".

To map TST design quality onto paradigm-specific design actions requires a *paradigm lens* through which design actions specific to the paradigm are interpretable in terms of TST design quality. A *paradigm lens* comprises: 1) a special ontology of that paradigm: individuals, attributes, relationships, and classes, and 2) design guidelines that associate the TST property design actions with corresponding actions in the specific paradigm. Where Thriving Systems Theory provides the lens onto design quality, the paradigm-specific special ontology and design guidelines provide the *paradigm lens* to focus on individual manipulations of paradigm-specific constructs to guide quality design choices.

The three courses in the design integration experiment share two paradigms: the relational and the object-oriented. The accompanying *paradigm lens* for each may be accessed in the literature (Waguespack, 2013 & 2015a). Lectures on Thriving Systems Theory and paradigm-specific ontology form the design pedagogy accompanied by design choices examples to illustrate choice properties. The courses employ the ontological terminology of the respective paradigms throughout. The design pedagogy continues to evolve with the experience of teacher and student.

8. DISCUSSION

This paper argues there is clear evidence that computing artifact design is a crucial professional competency and deserves concerted effort in IS and computing education. The special ontology and narrative of design delineate the fundamentals of the design process and highlight coursework opportunities. Where course credit hours are in short supply, adding design to a curriculum may (more often than not) require integrating design into existing courses. The integration scheme for these three courses demonstrates a feasible approach that is potentially elegant if carefully aligned with existing course objectives. Thriving Systems Theory and the paradigm-based special ontologies not only illuminate design quality but also, add theoretical depth to the pedagogy of the respective paradigms.

The special ontologies pose challenges for undergraduate students. However, feedback indicates that after the initial "shock" of studying object-orientation or relational modeling through ontology, later the "light goes on" when the same terminology clarifies the evaluations of their design products. Graduate students dosed twice with the TST design pedagogy report the pleasant surprise that the ontologies make team communication more clear and that TST design actions simplify their models.

Expanding this pedagogy approach to other topic areas is under consideration: agile development methodologies, virtual machine organization, and cloud based security architecture (Waguespack & Schiano, 2012; Waguespack, Yates & Schiano, 2014; Waguespack, 2014; Waguespack, Schiano & Yates, 2015).

9. REFERENCES

Alexander, C., Ishikawa, S. & Silverstien, M. (1977), *A pattern language: towns, buildings, construction*; Oxford University Press, London.

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

Alexander, C. (2002). *The nature of order an essay on the art of building and the nature of the universe:* Book I - The Phenomenon of Life, The Center for Environmental Structure, Berkeley, CA, 2002.

Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R.C., Mellor S., Schwaber K., Sutherland J., & Thomas D. (2001). *Manifesto for Agile Software Development*. Retrieved July 9, 2013 from agilemanifesto.org

Babb, J. S., Waguespack, L. J. (2014). In search of a design-focus in IS curricula, *Information Systems Education Journal, 12(4) pp 29-39. HTTP://isedj.org/2014-12/ ISSN: 1545-679.*

Brown, T. (2009). *Change by design*, HarperCollins, New York, NY.

Cohen, R. (2014). Design thinking: a unified framework, Forbes Online, www.forbes.com/ sites/reuvencohen/2014/03/31/design-thinking-a-unified-framework-for-innovation/ (accessed 7/6/2015).

Cross, Nigel (2007), Designerly ways of knowing, Birkhäuser Verlag AG Berlin.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995), *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, Boston.

Gero, J. S. (1990), Design prototypes: a knowledge representation schema for design, *AI Magazine*, 11(4), pp. 26-36.

Gero, J. S. & Kannengiesser, U. (2002), The situated function-behaviour-structure framework, *Artificial Intelligence in Design '02*, Kluwer Academic Publishers, Dordrecht, pp. 89-104.

Gregor, S. & Jones, D. (2007), The Anatomy of a Design Theory, *Journal of the Association for Information Systems*, 8(5), 312-335.

Hevner, A. and Chatterjee, S. (2010). *Design research in information systems: theory and practice*, Springer, New York, NY.

Medina, R. (2008). Christopher Alexander still matters. http://www.regismedina.com/blog/2008/04/ christopher-alexander-still-matters. Accessed June 15, 2013.

Mitchell, M.T. (2006). *Michael Polanyi*, Intercollegiate Studies Institute, Wilmington, DE.

Schön, D. (1983). *The reflective practitioner: How Professionals Think in Action*, Basic Books, New York, NY

Schön, D. (1983). *Educating the reflective practitioner*: Jossey-Bass, San Francisco, CA

Simon, H. (1996). *The* sciences *of the artificial*, 3rd Ed., M.I.T., Cambridge, MA

Turner, D. (2007), The secret of Apple design, MIT Technology Review, http://www.technologyreview.com/featuredstory/407782/the-secret-of-apple-design/, Accessed July 10, 2015.

Usher, R., & Bryant, I. (1989), *Adult education as theory, practice and research: The captive triangle*. Routledge, London.

Waguespack, L.J. (2010). *Thriving systems theory and metaphor-driven modeling*, Springer-Verlag, London.

Waguespack, L. (2011). Design, The "Straw" Missing From the "Bricks" of IS Curricula, *Information Systems Education Journal,* 9(2) pp 101-108. http://isedj.org/2011-9/ ISSN: 1545-679X.

Waguespack, L.J. & Schiano, W.T. (2012). SCRUM Project Architecture and Thriving Systems Theory. *45th Hawaii International Conference on System Science (HICSS)*, 4943-4951, 4-7 Jan.

Waguespack, L.J. & Schiano, W.T. (2013). Thriving Systems Theory: An Emergent Information Systems Design Theory. *46th Hawaii International Conference on System Science (HICSS)*, pp.3757, 3766, 7-10 Jan. 2013

Waguespack, L. (2013). A Design Quality Learning Unit in Relational Data Modeling Based on Thriving Systems Properties, *Information Systems Education Journal*, 11(4) pp. 18-30. http://isedj.org/2013-11/ ISSN: 1545-679X.

Waguespack, Leslie J., Yates, David J., Schiano, William T. (2014), Towards a Design Theory for Trustworthy Information Systems, 47th Hawaii International Conference on Systems Sciences, Hawaii, HI.

Waguespack, L. (2014) Computer Security Primer: Systems Architecture, Special Ontology and Cloud Virtual Machines. *Information Systems Education Journal*, 12(2) pp. 18-28. http://isedj.org/2014-12/ ISSN: 1545-679X.

Waguespack, L. (2015), A Design Quality Learning Unit in OO Modeling Bridging the Engineer and the Artist, *Information Systems Education Journal*, 13(1) pp. 58-70, http://isedj.org/2015-13/ ISSN: 1545-679X.

Waguespack, Leslie J., Schiano, William T., Yates, David J. (2015). "Trust = Utility + Security: Designing for that elusive quality in cloud computing," *The International Journal of Design in Society*, Champaign, IL (to appear)

Walls J.G, Widmeyer, G.R. & El Sawy, O.A. (2004). Assessing Information Systems Design Theory in Perspective: How Useful was our 1992 Initial Rendition?, *Journal of Information Technology Theory and Application*, 6(2), 43-58.

Walsham, G. (1993). *Interpreting information systems in organizations*. John Wiley & Sons, Inc., Hoboken, NJ