Position Paper

A Proposal for a Formal Definition of the Design Concept[1]

Yair Wand

Sauder School of Business
The University of British Columbia
Vancouver, BC, Canada, V6T 1Z2
yair.wand@ubc.ca

Abstract

It might appear surprising, but it seems no generally-accepted, precise and useful definition of what design is exists. The availability of such a definition would provide a clear reference for those engaged in design-related practice, research and education. In this paper, we suggest a definition for design. The definition views the design activity as a process whose purpose is to generate specifications of a given type of artifact based on four elements: definition of environment in which the artifact will be embedded, requirements for properties (structural or behavioral) of the artifact, a given set of possible component types, and constraints. For practical reasons we also include goals. We demonstrate the application of our definition and propose how it might be used in knowledge sharing, research, practice, and education. We hope this paper will lead to further discussion and examination of definitions of design.

Problem Statement: What is Design?

There are at least five reasons why a precise and accepted definition of what design is can be useful. First, from a pragmatic point of view, it can help organize, share and reuse design knowledge. Such sharing can enhance software project success and software development productivity. Second, to build cumulative knowledge via research on design phenomena, we need to agree on the common thread of these phenomena. This cannot be done without agreeing on what design is. Third, when evaluating different approaches to and methods for creating designs we need ways to evaluate the outcomes – namely the generated designs. This requires that these outcomes have some well-defined attributes that can be assessed. Fourth, when engaging in design practice, it would be useful to define the relevant issues to be considered and to be able to plan the process of design accordingly. Fifth, to be able to educate designers, we need to define the knowledge that needs to be included in educational programs.

Recently, there have been several calls for considering design as the subject for research, notably in the software and information systems domain. For example, Freeman and Hart (2004) call for a comprehensive, systematic research effort in the Science of Design. Hevner et al. (2004) argue that a "theory of design in information systems, of necessity, is in a constant state of [Kuhnian] scientific revolution," (p. 81). We claim that to study design and develop a credible theory of design in IS (or in any domain) requires a clear understanding of what design is. Many variables of interest, such as design project success, cannot be appropriately measured without such

---

[1] This position paper is based on work done with Paul Ralph – a PhD student at the Sauder School. A more detailed version appears in the *Proceedings of the Design Science Research in Information Systems and Technology Conference*, Pasadena, CA, May 2007. The work was supported in part by grants from the Natural Science and Engineering Research Council of Canada.

understanding. A clear and unambiguous definition of the design concept itself would at least be very useful, if not a necessary, condition for developing a common base for research on design.

Yet, although it might appear surprising, it seems no generally-accepted, precise and useful definition of what design is exists. The availability of such a definition would provide a clear reference for those engaged in design-related practice, research and education

What the Literature Says

It is not that no definitions of design exist in the literature. We have identified at least twenty-seven definitions of design and sub-types of design (such as "software design" and "urban design"). Though design has several meanings, we have focused on meaning involving plans for an object and planning or devising as a process.

Prior to comparing extant definitions, we evaluated them using four main criteria: coverage, meaningfulness, unambiguousness and ease of use. We adapted the first three criteria from evaluation criteria for good theories (e.g. Casti, 1989, p.44-45). The fourth criterion is a pragmatic one. We view the first three of these criteria as necessary, and the fourth as indicating usefulness. The criteria are:

(1) ***Coverage***: the definition should include all and only appropriate phenomena.

(2) ***Meaningfulness***: All terms and combinations of terms comprising a definition must either have a commonly accepted meaning in the given context or must have been pre-defined.

(3) ***Unambiguousness***: Each term used in a definition must have exactly one meaning in the given context.

(4) ***Usability:*** A definition should be easy to understand and remember, applicable in disparate situations, and readily differentiate between included and excluded phenomena.

We have not identified an article written specifically to define design. Moreover, often a definition of the concept was not included in documents related to design, such as the UML 2.0 specification. To demonstrate some difficulties we identified in extant definitions we provide two examples:

First, consider the definition provided in Engers et al. (2001): "the creative process of coming up with a well-structured model that optimizes technological constraints, given a specification." This definition seems to have both meaningfulness and coverage problems. The meaning of 'optimizes technological constraints' is unclear. One can optimizes an objective function subject to constraints, but not the constraints themselves. Second, the use of "well-structured" confounds the notion of design with that of design quality.

Second, Hinrichs (1992) defines design as "the task of generating descriptions of artifacts or processes in some domain," (p. 3). This definition seems to encompass phenomena one usually would not include in design studies. In particular, describing what exists is usually the purpose of analysis, not design.

Based on our subjective assessment all 27 definitions we have analyzed had coverage problems and about half included terms with unclear or ambiguous meaning. Eight definitions viewed design as a process, seven included the term planning, seven referred to design as a physical activity (or as including implementation), seven referred to design as creation, six mentioned the notion of system (as the subject of design) and five indicated or implied design was a deliberate, or purposeful activity.

Our Suggestion for a Definition of Design

We first observe that design can be referred to as an object (noun) or as an activity (verb). We also observe that there are different types of artifacts, not necessarily physical, or even symbolic, that can be designed. We call the designed artifact the ***subject***, and define:

***Design as an object*:** a specification of a *subject* intended to accomplish *goals*, in a particular *environment*, using a set of *primitive components*, satisfying a set of *requirements*, subject to some *constraints*.

***Design as an activity**:* to create a design (comprising the six elements above).

The key to the definitions is defining the five elements in addition to the subject, and to explain what is meant by "specification".

Environment: The context or scenario in which the subject is intended to exist or operate.
Examples: a school district (for an education system); an assembly plant (for an industrial robot).

Goals: The set of purposes the subject is supposed to achieve, in terms of its intended effects on the environment.

Requirements: a set of specific properties the subject needs to have. Partitioned into two groups: Functional and Non-Functional

Functional: the changes or the effects the subject should be able to make on its environment, in response to stimuli.
Example: a computer program will run a procedure when an icon is clicked.

Non-Functional: additional requirements which reflect some measures on the subject.
Example: a computer program is required to provide fast response.

It is important that goals are distinct from requirements. Goals reflect the intended effect or impact of the subject on its environment. Requirements are statements about the subject. Moreover, it is conceivable design can be undertaken based on requirements alone, without explicit knowledge of the goals. However, we contend goals should be included at least in the definition (even if omitted in practice) on pragmatic grounds to provide the context and rationale for the design.
Example: "increase worker productivity" is a goal for a new IS; "provide timely information" is a requirement.

Set of primitives: The set of types of elements (components) from which the subject may be composed.
Examples: {beams, nuts, bolts} for mechanical design; {programming language primitives, class library} for program design.

This set defines what the designer "has to work with." Often it is not provided explicitly but rather assumed. However, it might greatly affect the design process outcomes. For example, bridge design might vary considerably depending on the available materials.

Constraints: The set of behavioral and form restrictions on the subject.

Constraints are distinct from requirements. Requirements specify desired properties while constraints limit the possible solutions. Note the difference between non-functional requirements and constraints. Requirements refer to the subject while constraints will limit the acceptable specifications, even if they meet the requirements.

The definition of environment, goals, requirements, primitives, and constraints are inputs to the design activity while the specification is the outcome. However, in practice, not all inputs might be known in advance. For example, different sets of primitives might be explored as part of the design activity, and constraints may not be established at the outset. Moreover, the design process might uncover conflicts between different aspects. For example, a given set of primitives might not provide for certain requirements or cannot satisfy certain constraints.

Finally, we need to establish a clear meaning for the term "specification" which is the purpose of the design process. We suggest that specification is a detailed description of a subject in terms of the components used (out of the set of possible primitives) and their connections. Often the specification is provided in a symbolic representation (documents, diagrams). However, it does not necessarily have to be made explicit in such form. Sometimes the specification exists only in somebody's mind, presented as a physical model, or even embedded in the subject itself.

Finally, we have indicated the subject is the type of artifact to be designed. We have identified at least five types of artifacts:

- **Physical artifacts,** both simple, made of a single component, and composite, made of many components (which can be of different types)
- **Processes**, such as business workflows
- **Symbolic systems**, such as programming languages
- **Symbolic scripts**, such as essays, models and software
- **Laws, rules and policies**, such as a criminal code

For simple artifacts, the specification would include relevant properties (e.g. shape, weight and material). For composite artifacts it would include the components and their couplings. For processes that can be defined as 'a set of partially ordered activities aimed at reaching a goal' (Hammer and Champy, 1994)), a specification could identify the activities and their order. For a symbolic system, the specification can include syntax, semantics and pragmatics. A symbolic script can be specified by symbols used (from a symbolic system) and their arrangement. A rule can be specified in some (possibly formal) language (again using a symbolic system).

How the Proposed Definition Works

We demonstrate the use of the definition by applying it to two distinct domains.

Domain	Architecture	Software
Subject	An office buildings	An information system
Environment	An urban center	A given hardware and software platform
Goals	Provide office space	Support customer management
Requirements	Functional: Numbers of offices, meeting rooms, etc. Include open floor plan offices; Non-Functional: Be energy efficient	Functional: "maintain customer data" Non-Functional: "maintain security"
Primitives	Construction and interior design materials and components available	Hardware and software platforms and development environment available
Constraints	Limits on cost, height and total area.	Cannot use certain platforms (due to security considerations)

We now assess our definition with respect to six areas of agreement in the literature.

Design as a process: we define the design process as a set of steps resulting in a specification.
Design involves planning: in our definition planning is encapsulated by the design 'specification'. However, planning may be implicit, especially where specification occurs simultaneously with creating the subject.
Design as a physical activity (involving implementation): our definition is silent with respect to the nature of the activity. It does not include implementation, unless the implementation is the way of expressing the design.
Design as creation: we define the specific outcome that needs to be created.
Design involves a system: our definition does not relate to the concept of a system. It recognizes different types of artifacts that might not necessarily be viewed as systems (e.g. a set of rules).
Design is deliberate, i.e., has a purpose: we define the purpose of design as creating a specification for an artifact that has goals.

How the Definition can be used

At the outset of this paper we suggested there would be at least five areas where a clear definition of design can be useful. Here we refer to some of those.

Sharing Design Knowledge

The six elements- Subject, Goals, Environment, Primitives, Requirements and Constraints can be used for classifying and tagging standard designs for sharing purposes. To illustrate this use we provide two examples in the table below:

	iPod	Iterator
Subject	Physical artifact: consumer electronic	Symbolic script: Design Pattern
Environment	Entertainment consumer electronics market	Software systems (object-oriented)
Goals	Provide mobile auditory entertainment	provide a solution for accessing the elements of a collection of objects
Primitives	Available components (e.g. LCD small screens, batteries, miniature hard-drives, a selection of processors).	The primitives and class library of a given development environment.
Requirements	Functional: Store digital files; transfer files to and from a computer, play music files in several formats. Non-Functional: look attractive, given playing time per charge, compatibility with major operating systems.	Functional: have methods to return the next element in a collection and to check whether the collection empty; keep track of the current position in the collection Non-Functional: must be reasonably easy to understand.
Constraints	Fits easily in the hand, maximal weight, maximal dimensions.	Must not reveal details of underlying data structures; Must not violate rules of common object oriented languages.

By organizing design knowledge according to these dimensions, a design knowledge seeker could search for items such as 'consumer electronic devices (subject) for playing mobile video (goal)' or 'fast (constraint) algorithms (subject) for searching (goal).'

Implications for Research

We anticipate several ways the definition can be used in design research. First, on a general note, clear and consistent definitions of key terms can help the development of a cumulative body of research and support, for instance, meta-analysis of empirical results and integration of theories. Second, a clear definition of design is necessary for creating causal models of design quality, design project success, and for applying good design advice (e.g. Hevner et al., 2004). Third, the elements of the definition can suggest some research questions. For example, one can study the deficiencies that can arise if not enough attention is paid to constraints at the outset of design. Fourth, the identified elements of design can provide criteria for evaluating the quality of designs generated when comparing alternative methods.

Implications for Practice

The proposed definition suggests criteria for practical evaluation of designs and the resulting artifacts: Environment – can the artifact be used in the specified environment? Goals – does the artifact accomplish the expected outcomes? Requirements – is the artifact complete in providing for all requirements? Primitives – do we have all necessary elements to implement the artifact? Constraints- does the artifact meet all constraints? Answering to some of these questions not only can provide a measure of project success, but might even have legal implications.
Second, the breakdown of design into the elements provides a checklist for practitioners. Each element should be explicitly identified for a design task to be fully defined. A project manager might not be able to provide consistent and accurate estimates of a project costs if crucial elements are unknown. Third, a clear understanding of the difference between design and implementation can prevent poor decisions and evaluation practices.

Implications for Education of Designers

The definition of design can be used to educate designers in several ways. First, the definition provides a clear checklist of information a designer must seek prior to actually engaging in creating the required design. For each of the different aspects techniques can be practiced on how the information can be obtained. Second, it can enable trainees to search in design knowledge bases using the various elements as search keys.

Some Practical Considerations

If the definition is to be used in practice, several issues might arise, in particular with respect to information systems and software design. We address some of these here.

- The definition assumes a given set of primitive (component) types available to the designer. These components in themselves might be the subject of design. The designer might consider these components as available even they do not yet exist. This provides for *hierarchical design* where the properties of the assumed components can serve as requirements for their design.

- In software design, the specification and the subject may be the same. This will occur when the primitives are directly executable on some "machine" (real or virtual). In such cases, design and implementation are combined.

- Some of the requirements and constraints might be known prior to the design process while others might be implicit in the goals and environment and thus need to be revealed in the process.

- In the software domain, often software architects leave design decisions to the programmer; thus, programming also includes design activities.

- Recording the structure or properties of an existing artifact is not design. For example, if a business analyst creates a model of an existing business process, this would be considered analysis, not design.

Summary

In this paper we observe that a clear, precise and generally accepted definition of the concept of design is important for both research and practice of design. Our literature study indicates that such a definition is still not available. We therefore propose a definition for design. The definition views the design activity as a process whose purpose is to generate specifications of a given type of artifact based on four elements: definition of the environment in which the artifact will be embedded, requirements for properties (structural or behavioral), a given set of component types, and constraints. For practical reasons we also include goals.

We demonstrate the application of our definition and propose how it might be used in knowledge sharing, research, practice, and education. We hope this paper will lead to further discussion and examination of definitions of design.

References

Casti, J. (1989) *Paradigms lost*. Avion Books, New York, USA.

Engers, T., Gerrits, I., Boekenoogen, M., Glassee, I., & Kordelaar, P. (2001). POWER: Using UML/OCL for modeling legislation – An application report. ICAIL-2001, St. Louis, Missouri, USA.

Freeman P., & Hart, D. (2004) A Science of design for software-intensive systems. *Communications of the ACM*, 47 (8): 19-21.

Hammer, M. and Champy, J. (1994) *Reengineering the Corporation – A Manifesto for business revolution*, Nicholas Brealey Publishing, London.

Hevner, A., March, S., Park, J., & Ram S. (2004) Design science in information systems research. MIS Quarterly, 28 (1): 75-105.

Hinrichs, T. (1992) *Problem solving in open worlds: A case study in design*. Laurence Erlbaum.