Towards a framework of design principles: Classifying system features, behaviours and types

Chih-Chun Chen, University of Cambridge, Engineering Design Centre
Nathan Crilly, University of Cambridge, Engineering Design Centre

Abstract

‘Modularity’, ‘redundancy’, ‘robustness’, … these and other terms refer to principles that are well known in design research and widely applied in many varieties of design practice. What is less well considered within design is that these same principles are invoked by scientists as a way to characterise the structure, function and underlying ‘logic’ of biological systems. More generally, they are also being studied in a wide variety of disciplines concerned with defining, modifying or maintaining systems, whether those systems are comprised of hardware, software, ecologies, economies, societies or some combination of these. This widespread interest in ‘design principles’ and, in particular, their attention from biologists, provides an opportunity for design research to provide other disciplines with well defined, well characterised and well related concepts. However, in design, science and elsewhere, the lists of design principles offered are often developed in a seemingly *ad hoc* manner and are evidently (and knowingly) incomplete. This paper suggests that a framework can be developed which structures the existing design principles in a way that is applicable across different types of system. We explore the foundations upon which such a framework could be built by drawing on work from a broad range of disciplines.

Keywords

design; science; system architectures; biology; modularity; redundancy; ilities

In recent years, principles of design have been discussed in some of the world’s most celebrated scientific journals, such as *Nature* and *Science*. Here, scientists who are attempting to explain, model and engineer biological systems, invoke lists of design principles as a way to characterise the structure, function and underlying ‘logic’ of those systems. This has taken place seemingly without attracting much attention from design researchers and without drawing explicit contributions from the design research disciplines. With this paper we aim to bring these design-relevant aspects of contemporary scientific discourse to the attention of design researchers and to explore what design research might be able to offer to these ongoing discussions. More generally, the design principles at issue are applicable across a wide variety of systems and so gaining a better understanding of those principles, and how they are related to each other, would contribute to many different fields and also to design research itself. As such, we take the biological literature as a point of departure from which to explore the representation of design principles that are seemingly universal or at least widely considered.

We start with a short list of quotations from prominent scientific publications, each of which lists a set of design principles. These quotations have been selected to reveal the design-focus that is evident in the scientific discourse and also the connections and contrasts between the lists of design principles that are offered. For clarity, each quotation is accompanied by a brief explanation of its scientific context:

> "biological networks are seen to share structural principles with engineered networks. Here are three of the most important shared principles, modularity, robustness to component tolerances, and use of recurring circuit elements." (Alon, 2003, p. 1866)
>
> > - In biological networks, nodes represent molecules and edges represent biologically significant interactions between molecules (e.g. those that contribute to gene transcription regulation). Interactions can vary in both strength and valence.

> "The notion of a 'part' [as used by biologists] is essentially an engineering concept, reflecting important synthetic goals of modularity, standardized structural and functional composition, hierarchical assembly, isolation from other components, characterized behavior, and standardized interfaces." (Knight, 2005)
>
> > - Synthetic biology is research field and a technology application relating to the design and construction of biological devices and systems. Knight's discussion relates primarily to genes, where the 'part' is a snippet of DNA with a defined biological function.

> "Five popular concepts in biology today — redundancy, robustness, modularity, complexity and evolvability — invoke a vision of the cell as an electronic circuit, designed by and for adaptation." (Lynch, 2007, p. 803)
>
> > - This discussion is centred on the evolution of genetic networks and the mechanisms by which they become established evolutionarily.

These quotations provide lists of what are variously referred to as "principles" or "concepts" but are hereafter simply called *principles*. Each of the principles in these lists represents a particular way of thinking about the system in question, a way of understanding its organisation and the effect of that organisation on its performance. Although the scientific literature cited above makes very limited reference to the design literature, the principles listed are well discussed in design and in related literatures such as engineering and systems architecture. For example, Baldwin & Clark's (2000) five-hundred-page book titled *Design Rules* is devoted to the concept of modularity in the design, manufacture and use of products (also see e.g. Gershenson, Prasad, & Zhang, 2003, 2004; Ulrich, 1994). Attention has similarly been devoted to issues of designing with redundancy (e.g. Pahl & Beitz, 1996; Thompson, 1999), robust design (e.g. Taguchi & Clausing, 1990; Downey, Parkinson, & Chase, 2003), complexity engineering (e.g. Buchli & Santini, 2005; Frei & Serugendo, 2011a, 2011b), and so on. In these texts and elsewhere, researchers define these principles, differentiate them from other principles, identify sub-principles and explore the basis upon which the principles should be applied and the consequences of applying them. The principles themselves can be seen at work in the practice of engineering design, industrial design, software design, service design and a host of other design disciplines. For example, we can identify redundancy in the architectures of aircraft propulsion systems, computer storage devices, human communication systems, and in many other types of artefact.

If referring to design principles is an effective way for scientists to think about the structure and function of biological systems, then developing more refined lists or other categorisations would offer a better foundation for that work. Considering the quoted lists in this way suggests a number of questions we might ask when undertaking such an exploration:

1. *Are the items in the lists all the same type of thing?* Note that saying they are all 'principles' is still quite vague; some items are descriptions of structure whereas some are descriptions of behaviour.
2. *Are the lists complete?* Note that each list contains items missing from the other lists but then it is not clear whether they are all listing the same type of thing.

3. *Are simple lists the most appropriate form of representation for these items?* Note that if the relationship between the items is complicated then lists are generally inadequate.

To address these questions, this paper reviews the literature from engineering design, systems engineering and elsewhere that explores how we might identify and categorise the fundamental design principles that apply to technical and biological systems. As we shall see later, these principles also extend beyond technology and biology, with application to ecologies, organisations, economies, and other systems that we might want to specify, modify, understand or protect.

Before, proceeding with the identification and classification of design principles, it is important to distinguish this work from a related project with which it might be confused. In recent years, various fields of design have looked to biology for inspiration in solving technical problems; this is often called *biomimetics* (but is also called *bionics* or *nature-inspired design*). In general, these approaches look for biologically-inspired solutions to quite specific technical problems. For example, The Biomimicary 3.8 Institute (2008) categorises biological systems not on the basis of groups such as genus or species, but on the basis of functional requirements such as adhesion, locomotion and lubrication. This categorisation of biological solutions can then be used by designers to identify analogous solutions to technical problems, e.g. problems of reversible adhesion, robust locomotion or self-lubrication.

Although system architectures are seldom the focus of biomimetic approaches, there are some authors who describe the architecture of technical systems as though those architectures are founded on biological principles. For example, in considering the design of complex computing environments, Forrest et al. (2005, p. 208) say that “Among the principles of living systems we see as most important to the development of robust software systems are: modularity, autonomy, redundancy, adaptability, distribution, diversity, and use of disposable components.” Some of these principles are identical to those offered by biologists but Forrest and colleagues are suggesting that biological systems exhibit principles that are important to the design of technical systems. They thus make the opposite argument to that made in the opening quotations. As such, in this paper, we do not take the view that biology borrows concepts from technology or vice versa, but just that there are fundamental design principles that apply across both domains and also across other domains.

Before discussing design principles and their classification, it is worth briefly considering what classification is and how it works. This is important because in grouping principles together and distinguishing them from each other we need to be explicit about the basis upon which commonalities and differences are observed. This helps not only in the identification of principles, but also in how we decide to relate them to each other and how we represent them.

Classification

Any attempt to provide a list or framework of design principles involves engaging in a basic process of classification. As we saw in the quotations that open this article, some of the existing lists in the science literature mix different types of thing together without comment, and if this is to be improved upon, some attention to the principles of classification is warranted. Bailey defines classification as “the ordering of entities into groups or classes on the basis of their similarity” (Bailey, 1994, p. 1). However, as Marradi observes, any entity can result in many classification schemes depending on the ‘classificatory principle’ used to determine similarity or difference: “Take for example the concept of a political system. One of its properties is the principle of legitimizing rulers. If that property is taken as *fundamentum divisionis*, the classes might be: theocratic, autocratic, plutocratic, democratic, etc. However, if the degree of autonomy of a State's territorial components is taken as the *fundamentum divisionis*, then the classes will probably be: unitary, federal, confederal, etc." (Marradi, 1990, p. 123). This variety of possible classifications for any set of entities means that it is important to be clear over the purpose of a classificatory scheme.

Along with other classification theorists (see Bowker & Leigh Star, 1999; Marradi, 1990), Bailey promotes classification based on two rules: (1) that the classes formed should be both *exhaustive* (i.e. everything is classified); and (2) that the classes should be *mutually exclusive* (things only appear in one class). It is also worth distinguishing between conceptually derived classifications and empirically derived classifications. Bailey (1994, pp. 4–6) calls these typologies and taxonomies, respectively, but terminology varies between authors (e.g. see Spradley, 1980, p. 115). Irrespective of the classificatory structure that is arrived at, there are at least two approaches to classifying things: (1) when proceeding from the general to the particular we can take an overall category (e.g. Dogs) and identify the various members of that category (e.g. Dalmatians, Poodles, Greyhounds, etc.); (2) when proceeding from the particular to the general we can take individual items (e.g. Dogs, Cats, People, etc.) and identify them as members of an overall category (e.g. Mammals). Often, both approaches are applied at the same time; we start with some overall categories and some individual items; by exploring the relationship between them we find that the general categories suggest the existence of individual items that are missing and also that individual items suggest the existence of general categories that are missing.

A cross-domain framework

If scientists are looking to define, characterise and classify the design principles that can be used to modify the behaviour of systems, what can design research offer in response? Attention to the design literature and to literature from other related areas suggests that we can identify opportunities for a framework of design principles rather than just a list of them. Rigorously developing a comprehensive framework of design principles is beyond the scope of this paper as it will require sustained effort (both conceptual and empirical) across a number of fields. However, by drawing together the various works that might contribute to such a framework, we aim to suggest the direction in which development might proceed and sketch the types of outcome that might be arrived at. The structure of our proposed framework is initially based on distinguishing system *features* from system *behaviours*. We consider features to be descriptive of a system's architecture or implementation (e.g. modularity and redundancy) and behaviours to describe how that system then permits change or responds to it (e.g. robustness and evolvability). There is prior work to assist us in constructing this framework, and we review this first before considering the features and behaviours separately in more detail. Finally we consider the way in which these features or behaviours are identifiable or applicable across different types of system and how those system types might be considered.

Two existing Frameworks

There are two frameworks in the systems engineering literature that divide design principles into classes that are similar to the features and behaviours distinction outlined above. Hastings and McManus (2004) are concerned about the ways in which uncertainties present risks and opportunities for complex systems, and how those can be mitigated or exploited to achieve desired outcomes. These last two issues, 'mitigations/exploitations' and 'outcomes', are of most interest to us here and so these aspects of their framework are reproduced in Figure 1. Writing of their framework, Hastings and McManus say “Completeness is not attempted, although the elements should be sufficient for a discussion of issues facing complex space systems and systems-of systems” (Hastings & McManus, 2004, p. 2). The elements are presented here on the assumption that they apply more widely than that.

Mitigations/Exploitations

Margins
Redundancy
Design Choices
Verification and Test
Generality
Upgradeability
Modularity
Tradespace Exploration
Portfolios & Real Options

Outcomes

Reliability
Robustness
Versatility
Flexibility
Evolvability
Interoperability

Figure 1. Two lists of design principles derived from the framework of Hastings and McManus (2004, p. 2).

In a related piece of work, Fricke and Schulz (2005) develop a framework that identifies four ‘aspects’ of changeability (similar to the ‘outcomes’ listed above) and contrast these with a set of ‘principles’ that support these aspects. Fricke and Schulz distinguish between Basic Principles and Extending Principles, where the former support all aspects of changeability and the latter support only selected aspects of changeability. The ‘aspects’ and ‘principles’ are reproduced in Figure 2 (but it should be noted that Fricke and Schulz integrate these into a more sophisticated diagrammatic form).

Principles

Ideality/Simplicity
Independence
Modularity/Encapsulation
Integrability
Autonomy
Scalability
Non-Hierarchical Integration
Decentralisation
Redundancy

Aspects

Flexibility
Agility
Robustness
Adaptability

Figure 2. Two lists of design principles derived from the framework of Fricke and Schulz (2005, p. 348).

Taken as they are, the two frameworks described above each provide a very valuable response to the quotations that open this article. However, by comparing these two frameworks we can explore the matter further. At first glance, it appears that Hastings & McManus and Fricke & Schulz provide similar lists, and indeed this is the case. Loosely speaking, both frameworks list a range of features that designers can seek to implement in the architectures of the systems they design (the ‘mitigations/exploitations’ and ‘principles’), and also list a range of behaviours that these features promote or support (the ‘outcomes’ and ‘aspects’). Also, what is common to both frameworks and to the quotations that open this paper is the system features of *modularity* and *redundancy* and the system behaviour of *robustness* (*flexibility* is also mentioned in both frameworks). However, the rest of the items in each framework are unique to that framework, or put another way, are missing from the other framework. So, there is good agreement over the structure of the frameworks (the basic division between lists of features and lists of behaviours) but only limited agreement over their content (only a small number of items in those lists are common).

Considering the frameworks above yields the following observations. First, *flexibility* is presented as a feature in one of the frameworks but as a behaviour in the other. Second, *scalability* and *upgradability* are presented as features even though they seem to describe the same type of thing as the behaviours of *flexibility*, *agility* and *adaptability*. Taken together, these observations might lead us to question the foundations of a feature-behaviour distinction, especially as the opposite observations can be made in other literatures, where *modularity* (here considered as a feature) is listed as a behaviour (de Weck et. al., 2012). The basis upon which features and behaviours might be distinguished from each other clearly deserves scrutiny, but we proceed on the assumption that the basis is sound whilst acknowledging that further work is required. Finally, it can be observed that some of the listed features might better be regarded as design and development methods: *verification and test*, *tradespace exploration*. Such items might instead be separated into another list altogether, a list of the methods by which the system features are decided upon and implemented.

System features

Although the two-list structure promoted in the frameworks above is not often found in the literature, much attention has been given to individual items in each list. Across various literatures, some of the most commonly cited design principles are *modularity* and *redundancy*. These two principles have been given a relatively thorough treatment in the design literature, but their formulation and classification tend to be system- or domain-specific and rather *ad hoc*. References to these principles are made in the quotations that open this paper. They also appear in the two frameworks proposed respectively by Hastings & McManus and Fricke & Schulz, as well as being widely cited elsewhere (Jen, 2005, p. 2; Lidwell et. al., 2010, pp. 160, 204). The near ubiquity of *modularity* and *redundancy* in lists of design principles demonstrates their widespread importance. They are also principles that have received a great deal of attention individually. In their respective literatures, m*odularity* and *redundancy* are sometimes treated as umbrella terms under which a number of sub-principles can be identified. For example, several varieties of modularity have been suggested (Ulrich & Tung, 1991; Ulrich, 1995; Huang & Kusiak, 1998; Arnheiter & Harren, 2005) as have several varieties of redundancy (Johnson, 1989; Pahl & Beitz, 1996; Wada et. al., 2006). These varieties are listed in figure 3.

Modularity

- Component Sharing, Component Swapping, Fabricate-to-Fit, Mix Modularity, Bus Modularity, Sectional Modularity (Ulrich & Tung, 1991)
- Integral architectures *vs* slot, bus and sectional modularity. (Ulrich, 1995)
- Function-oriented modules, production-oriented modules. (Huang & Kusiak, 1998)
- Manufacturing modularity, Product use modularity, limited life modularity, data access modularity. (Arnheiter & Harren, 2005)

Redundancy

- Passive hardware redundancy, active hardware redundancy, information redundancy, time redundancy, software redundancy (Johnson, 1989)
- Active, passive (Pahl & Beitz, 1996)
- Parallel, series (Pahl & Beitz, 1996)
- Principle, selective, comparative (Pahl & Beitz, 1996)
- Cold standby, hot standby, warm standby (Wada et. al., 2006)

Figure 3. Classifications of modularity and redundancy from the engineering literature.

The many classifications of *modularity* and *redundancy* suggests that simple lists might not always be sufficient for representing the design principles that are specific to a particular application. For example, Ulrich and Tung (1991) identify different types of modularity that are defined based on the architecture that results from different combinations of interface compatibility and assembly, while Huang and Kusiak (1998) focus instead on the purpose served by the modules. Similarly, the classifications defined in (Pahl and Beitz, 1996) seem to be concerned with different dimensions of redundancy. Serial and parallel redundancy are distinguished on the basis of whether the redundant components are sharing the load of realising the function during normal operation (parallel) or not (serial), while principle, selective and comparative redundancy are distinguished by the mechanism for realising redundancy (in principle redundancy, multiple non-identical components with different dependencies are all capable of fulfilling the function by different means so that systemic failure of all of them is highly improbable; in selective redundancy, many components fulfil the function in parallel but only one of the outputs is actually used; in comparative redundancy, comparisons between outputs permit confidence in the output). The fact that these classifications are not always mutually exclusive suggests that as well as considering modularity and redundancy as general design principles, some kind of hierarchy may be required that relates these more general principles to more specific sub-principles which can be applied in different areas of design practice.

Looking beyond *modularity* and *redundancy*, other system features can also be identified. Considering the 'organisational principles' that characterise various highly robust entities, Jen proposes a list that includes redundancy and modularity, but also "spatial structure, … diversification, and hierarchy among others" (Jen, 2005, p. 2). As with the more specific classifications within modularity and redundancy referenced above however, it is difficult to be exhaustive when defining such a list, and different organisational principles are likely to overlap or be subsumed under others. Identifying the classification that is most useful in a given scenario requires an intricate understanding of the requirements of the systems or products concerned.

System behaviours

Whilst principles such as modularity and redundancy describe features of a system's design or architecture, the quotations that open this paper also referred to principles such as robustness and evolvability. These are not features of a system of the sort discussed above, but are instead descriptions of the way in which a system will behave in the presence of change. From the two frameworks above, it can be seen that these behaviours are often described using words that end in '-ility', including *flexibility* (Fricke & Schulz, 2005; Hastings & McManus, 2004), *reliability*, *versatility, evolvability*, *interoperability* (Hastings & McManus, 2004), *agility* and *adaptability* (Fricke & Schulz, 2005). As a consequence of this recurring suffix, these behaviours are sometimes called 'the ilities' (although this term often refers to a broader set of concepts including quality and safety). This name persists even though other change-relevant system behaviours are described with very different words, including *robustness* (Fricke & Schulz, 2005; Hastings & McManus, 2004) and *resilience* (Pavard et al., 2006). Other collective names for the ilites are less common but more descriptive, including 'system lifecycle properties' (Ross, Beesemyer, & Rhodes, 2012) and sometimes 'non-functional requirements' (Glinz, 2007). Recent work has advanced our understanding of these system behaviours, considering their definition and differentiation (e.g. McManus, Richards, Ross, & Hastings, 2007; Ross, Rhodes, & Hastings, 2008; de Weck, Roos, & Magee, 2011). Depending on the authors' focus, many different lists of ilities have been proposed, but for our purposes here, those in Figure 4 are indicative of their kind and variety.

adaptability
agility
changeability
evolvability
extensibility
flexibility
interoperability
modifiability
modularity
reconfigurability
robustness
scalability
survivability

Figure 4. A list of ilities from (de Weck, Ross & Rhodes, 2012). Of immediate note in the list above is the presence of modularity, which we had previously considered as a system feature rather than a behaviour. But this is discussed further below.

Beyond collection and definition, work has been done to relate the ilities to each other, understanding their interdependencies and possible subcategories. Early attempts of this kind can be seen in Boehm et al.'s (1976, p. 595) conceptual classification of the determinants of software quality. More recent work reported by de Weck et al. (2011) used a keyword correlation analysis of academic articles to understand how peripheral ilities support the more central ones: for example, flexibility contains evolvability, adaptability, agility, scalability and extensibility, whilst modularity is a promoter or pre-requisite for these. Resilience was also deemed to be an umbrella term that contains agility, adaptability and parts of robustness (de Weck et al., 2011, pp. 84–90). In another study, a preliminary investigation of a possible means-ends hierarchy for ilities was conducted by having different groups of participants cluster fifteen ilities (de Weck, Ross, & Rhodes, 2012). Although the groups arrived at very different hierarchies there was some agreement about which were the high-level ilities (the *ends*: robustness, changeability) and which were the low level ilities (the *means*: modularity, interoperability). This broadly corresponded with the results from the keyword analysis of academic articles reported above and suggests that considering “sets” of ilities may be more meaningful than considering them in isolation.

Sets of ilities may be derived from grouping pre-defined ilities that have emerged from practice (as above), or by formally deriving a set of ilities that are distinguished from each other by specific components of their definition, for example, by distinguishing whether the source of system change lies inside or outside its system boundary (Ross et al., 2012). This latter approach might be used to define general ilities, before defining their sub-ilities (rather than identifying general ilities from clusters of existing sub-ilities). A simple example of this would be to distinguish systems that *can change* from systems that can *accommodate* change. Identifying the source of change and the type of change that apply to each would then permit the systematic definition of various sub-ilites. Either way, this discussion of umbrella ilites, sub-ilites and sets of ilities supports the point made earlier: some form of hierarchy may be useful when representing the known design principles, not just the system features but also the system behaviours. These hierarchies might explain the double-listing of modularity as both a feature and a behaviour, but the features-behaviours distinction seems to emerge from an analysis of the principles (whether termed means-ends, principles-aspects or mitigations/exploitations-outcomes).

System types

Most of the time, design principles relating architectural features to system behaviours (or ‘ilities’) are formulated within a particular application domain. For example, resilience (sometimes conflated with robustness and other ilities) is discussed specifically with respect to resilient organisations (Sheffi & Rice, 2005), cities (Campanella, 2006), ecologies (Folke et. Al., 2010; Holling, 1973) and a broad range of technical and socio-technical systems (Chun, Zhao & Kubiatowicz, 2005; Jackson, 2006; Pavard et al., 2006). Other authors suggest that such design principles apply not just within but across different types of system. For example, with respect to modularity, Schilling writes that “as a more refined theory of modular systems evolves, it should prove to be a very powerful instrument for understanding the integration and disaggregation of many kinds of systems, including organisational, technological, social, and biological systems.” (Schilling, 2000, p. 313). Similarly, writing of flexibility, agility, robustness, and adaptability, Fricke and Schulz say that “Although examples presented in this paper are mainly covering product systems, the authors are certain that the proposed principles are applicable to any type of system (e.g. processes, organisations, etc.)” (Fricke & Schulz, p. 346). Such claims would seem warranted, but have not been followed by a focussed examination of how the design principles are similar or different across domains.

Explicitly considering the application of design principles across different types of system would be aided by the availability of different classifications of system types, yet few widely used classification schemes adhere strictly to a consistent set of classificatory principles for each level. For example, the International Patent Classification (IPC) groups products and processes using diverse and multiple criteria (WIPO, 2013). Even at the top ‘Section’ level, there are classifications based on a diverse set of principles, including function (e.g. Section A - Human Necessities, Section B - Performing Operations, Transporting); substrate (e.g. Section D - Textiles, Paper; Section H - Electricity); domains or practices (e.g. Section C - Chemistry, Metallurgy, section G - Physics); and combinations of these (e.g. Section F includes Mechanical Engineering, Lighting, Heating, Weapons, Blasting). New categories are also constantly being added to accommodate emerging technologies. The difficulty of specifying these new categories has been explicitly noted by the European Patent Office (EPO), which acknowledges (with respect to nanotechnology), “Its interdisciplinary nature means that the literature – especially patent documents – is difficult for searchers to retrieve. The EPO has therefore introduced ‘Y01N’ tags to label nanotechnology in EPO databases. Recently Y01N became B82Y” (EPO, 2013).

The *ad hoc* nature of many commonly used classification schemes such as the IPC is likely to be a consequence of the way in which they were constructed, namely through the objects that need to be classified (the IPC is updated regularly to accommodate new inventions without having to reclassify previous inventions). However, this does not preclude classifications being developed with a top-down approach. Crawley et al. (2004) provide one such example, with their classification scheme for “things with architectures”. They first distinguish between entities and non-entities which inform or constrain entity architectures and then further classify entities according to whether they are natural or designed by people. For those entities that are designed by people, there are further sub-classifications (see Figure 5).

Crawley et al.’s classification provides a useful foundation for considering the various system types to which design principles might apply. However, the emergence of new technologies challenges distinctions that were previously thought to be obvious and clear cut. For example, biological systems can be both natural and synthetic, or they may have both natural and synthetic components. Therefore, for design purposes, rather than simply classifying systems, it may in many cases be more useful to classify the different perspectives we can take on a system. Rather than say that system *A* is a biological system and system *B* is not, it could be more productive to say that we are concerned with the biological aspect of system *A*, or that system *A* presents more biologically challenging issues than system *B*. Similarly we might say that system A is more natural or less synthetic than system *B* and hence that more of the design issues we need to address for system *A* are those that apply to natural systems (compared to system *B*).

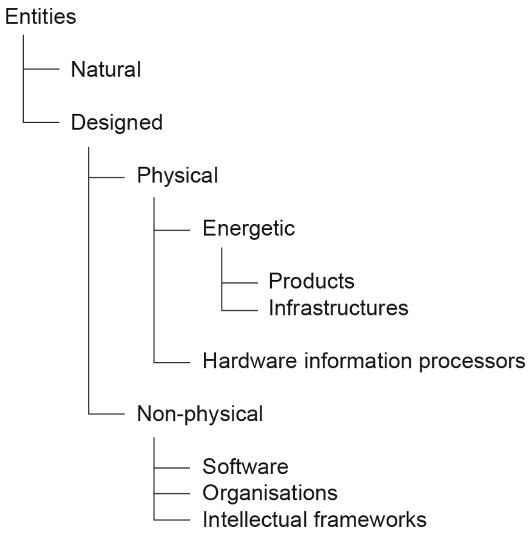

Figure 5: Classification of system types based on (Crawley et al., 2004, p. 15).

Different system types can be used as models and metaphors when designing or attempting to modify a system that would not in the first instance be intuitively characterised as being of that type (e.g. biological systems can be treated as social systems and vice versa, see Hemelrijk et al., 2011; both biological and social systems can be characterised in network terms, see Bonabeau et al., 1999; Dorogovtsev et al., 2003). Conversely, for some purposes, it may be highly beneficial to apply more fundamental design and engineering practices to systems that at first sight appear to be too complex to treat in this way. For example, Endy (2005) asks, “[c]ould we usefully consider adapting or extending ideas from structural engineering to synthetic biology?’ while at the same time acknowledging that “…evolution, is largely unaddressed within past engineering experience” (Endy, 2005, p. 450).

To date, little work has been done to assess the respective benefits and limitations of different classification schemes to promote cross-domain learning and innovation. For example, how does seeing a system in social terms or knowing that it developed through natural processes help us when designing or modifying it? Possible candidates for defining different system types (or different perspectives on systems) include the type of components involved (e.g. biological, structural, informational, mechanical, social), the type of interactions between components (e.g. consensus, logic, physical laws), and the type of process by which the system has been developed (e.g. designed/planned, evolved/emergent). Such classificatory principles can also be seen as defining different levels of abstraction because one could in theory view all systems as physical. The choice of classification may ultimately involve practical choices over the knowledge available, the accuracy of viewing systems in particular ways and the efficiency of doing so (Dennett, 1987). A characterisation scheme that would be useful in the design context should make explicit the benefits of using the scheme in terms of the existing design practices it allows us to exploit.

Discussion

The quotations at the beginning of this paper showed that scientists draw on design principles when they are trying to understand the structure and function of biological systems. Comparing the statements revealed significant variety in which principles were cited, indicating different areas of interest and the lack of a common source to draw from. From our survey of the relevant design literature, we identified a range of other principles to be considered and suggested some approaches to classifying them, including multiple lists based on different classificatory principles and hierarchies relating taxonomies and typologies defined at different levels. To better understand the formulation of design principles, we drew on a number of more specific literatures that detail the features of systems and the behaviours that those features promote.

From this initial starting point, we plan to conduct a more detailed analysis of the existing literature, broadening our scope to include disciplines that focus on understanding the composition and behaviour of complex systems. This will permit fieldwork with experts to understand how system features are related to each other for different system types. Such work will offer a better differentiated account of the design principles that govern the structure and function of systems generally, not just biological systems and technical systems.

A well developed framework of design principles holds implications for a variety of design practices. Taking a systems approach to technologies, organisms, ecologies, organisations, economies and other entities broadens the application of design principles and provides new perspectives that can push forward the development of design research itself. This is because each new system type that is examined is developed by different processes, is amenable to different interventions and responds to those interventions in new ways. As such, further attention to these design principles promises benefits for design research and for design research’s capacity to serve science and other disciplines.

Acknowledgments

This work was funded by the UK’s Engineering and Physical Sciences Research Council (EP/K008196/1).

References

Abernathy, W. J., & Utterback, J. M. (1978). Patterns of industrial innovation. *Technology Review*, 40–47.

Alon, U. (2003). Biological Networks: The Tinkerer as an Engineer. *Science*, *301*(5641), 1866 –1867.

Arnheiter, E. D. & Harren, H. (2005). A typology to unleash the potential of modularity. *Journal of Manufacturing Technology Management*, *16(7/8)*, 699-711.

Bailey, K. D. (1994). Typologies and taxonomies: an introduction to classification techniques. Thousand Oaks, CA: Sage.

Baldwin, C. Y., & Clark, K. B. (2000). *Design Rules: The power of modularity*. Cambridge, MA: MIT Press.

Boehm, B. W., Brown, J. R., & Lipow, M. (1976). Quantitative evaluation of software quality. In *ICSE ’76 Proceedings of the 2nd international conference on Software engineering* (pp. 592–605).

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999) Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press.

Bowker, G. C., & Leigh Star, S. (1999). *Sorting things out: classification and its consequences*. Cambridge, MA: The MIT Press.

Buchli, J., & Santini, C. C. (2005). Complexity engineering, harnessing emergent phenomena as opportunities for engineering. *Reports of the Santa Fe Institute's Complex Systems Summer School*.

Campanella, T. J. (2006). Urban Resilience and the Recovery of New Orleans. *Journal of the American Planning Association*, *72*(2), 141–146.

Chun, B-G., Zhao, B. Y., & Kubiatowicz, J. (2005). Impact of neighbour selection on performance and resilience of structured P2P networks. *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*.

Crawley, E., de Weck, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D., & Whitney, D. (2004). *The influence of architecture in engineering systems* (Engineering Systems Monograph). Engineering Systems Division, MIT.

Dennett, D. C. (1987). *The intentional stance*. Cambridge, MA: The MIT Press.

de Weck, O. L., Roos, D., & Magee, C. L. (2011). Engineering Systems: Meeting Human Needs in a Complex Technological World. MIT Press.

de Weck, O. L., Ross, A. M., & Rhodes, D. H. (2012). Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities). In *3rd International Engineering Systems Symposium*. Delft, The Netherlands.

Dorogovtsev, S. N. & Mendes, J. F. F. (2003). Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press.

Downey, K., Parkinson, A., & Chase, K. (2003). An introduction to smart assemblies for robust design. *Research in Engineering Design*, *14*(4), 236–246.

Ebeling, C. E. (1997). An introduction to reliability maintainability engineering. McGraw-Hill.

Endy, D. (2005). Foundations for engineering biology. *Nature*, *438*(7067), 449–453.

Folke, C., Carpenter, S. R., Walker, B., Scheffer, M., Chapin, T., & Rockström, J. (2010). Resilience thinking: integrating resilience, adaptability and transformability. *Ecology and Society*, *15*(4), 20.

EPO - European Patent Office (2013). Patent classification. http://www.epo.org/news-issues/issues/classification.html. Retrieved on 29/10/2013.

Forrest, S., Balthrop, J., Glickman, M., & Ackley, D. (2005). Computation in the wild. In E. Jen (Ed.), *Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies* (pp. 207–230). Oxford University Press.

Frei, R., & Serugendo, G. D. M. (2011a). Advances in complexity engineering. *International Journal of Bio-Inspired Computation*, *3*(4), 199–212.

Frei, R., & Serugendo, G. D. M. (2011b). Concepts in complexity engineering. *International Journal of Bio-Inspired Computation*, *3*(2), 123–139.

Fricke, E., & Schulz, A. P. (2005). Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. *Systems Engineering*, *8*(4), 342–359.

Gershenson, J. K., Prasad, G. J., & Zhang, Y. (2003). Product modularity: Definitions and benefits. *Journal of Engineering Design*, *14*(3), 295–313.

Gershenson, J. K., Prasad, G. J., & Zhang, Y. (2004). Product modularity: measures and design methods. *Journal of Engineering Design*, *15*(1), 33–51.

Glinz, M. (2007). On Non-Functional Requirements (pp. 21–26). Presented at the 15th IEEE International Requirements Engineering Conference.

Hastings, D., & McManus, H. (2004). A framework for understanding uncertainty and its mitigation and exploitation in complex systems. Presented at the Engineering Systems Symposium, MIT, Cambridge, MA.

Hemelrikj, C. (ed.) (2011). Self-Organization and Evolution of Biological and Social Systems. Cambridge University Press.

Holling, C. S. (1973). Resilience and Stability of Ecological Systems. *Annual Review of Ecology and Systematics*, *4*, 1–23.

Huang, C-C., Kusiak, A. (1998). Modularity in Design of Products and Systems. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 28(1)*, 66-77.

Jackson, S. (2006). A multidisciplinary framework for resilience to disasters and disruptions. *Journal of Integrated Design and Process Science*, 11(2), 91-1-8. Society for Design and Process Science.

Jen, E. (2005). Introduction. In E. Jen (Ed.), *Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies* (pp. 1–6). Oxford University Press.

Johnson, B. W. (1989). The Design and Analysis of Fault Tolerant Digital Systems. Addison-Wesley.

Knight, T. F. (2005). Engineering novel life. *Molecular Systems Biology*, *1*(1).

Lidwell, W. Holden, K., & Butler, J. (2010). Universal Principles of Design: 115 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions and Teach Through Design. Rockport, London.

Lynch, M. (2007). The evolution of genetic networks by non-adaptive processes. *Nature Review Genetics*, *8*(10), 803–813.

Marradi, A. (1990). Classification, typology, taxonomy. *Quality & Quantity*, *24*(2), 129–157.

McManus, H., Richards, M., Ross, A., & Hastings, D. (2007). A Framework for Incorporating “ilities” in Tradespace Studies. In *AIAA Space* (Vol. 1, pp. 941–954).

Pahl, G., & Beitz, W. (1996). *Engineering design: a systematic approach*. London, UK: Springer.

Pavard, B., Dugdale, J., Saoud, N. B., Darcy, S., & Salembier, P. (2006). Design of robust socio-technical systems. In *In Proceedings of 2nd Resilience Engineering Symposium*. Ecole de Mines de Paris, France.

Ross, A. M., Beesemyer, J. C., Jr, & Rhodes, D. H. (2012). *A Prescriptive Semantic Basis for System Lifecycle Properties* (Working paper). MIT, Cambridge, MA.

Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2008). Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. *Systems Engineering*, *11*(3), 246–262.

Schilling, M. A. (2000). Toward a general modular systems theory and its application to interfirm product modularity. *Academy of management review*, *25*(2), 312–334.

Sheffi, Y., & Rice, J. B. (2005). A supply chain view of the resilient enterprise. *MIT Sloan Review*, Fall.

Spradley, J. P. (James P. (1980). *Participant observation*. New York, NY: Holt, Rinehart and Winston.

Stellman, A., & Greene, J. (2008). *Applied Software Project Management*. O’Reilly Media.

Taguchi, G., & Clausing, D. (1990). Robust Quality. *Harvard Business Review*, *January-February*, 65–75.

The Biomimicry 3.8 Institute. (2008). AskNature. Retrieved August 7, 2013, from http://www.asknature.org/

Thompson, G. G. (1999). Improving maintainability and reliability through design. Professional Engineering Publishing.

Ulrich, K., & Tung, K. (1991). Fundamentals of Product Modularity. In *Winter Annual Meeting Symposium on Issues in Design Manufacture/Integration*, *ASME*, 73-79.

Ulrich, K. (1994). Fundamentals of Product Modularity. In S. Dasu & C. Eastman (Eds.), *Management of Design* (pp. 219–231). Springer, Netherlands.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research policy*, *24*(3), 419–440.

Vincent, J. F. V., Bogatyreva, O. A., Bogatyrev, N. R., Bowyer, A., & Pahl, A.-K. (2006). Biomimetics: its practice and theory. *Journal of The Royal Society Interface*, *3*(9), 471–482.

Wada, H., Suzuki, K., Oba, K. (2006). Modeling Non-Functional Aspects in Service Oriented Architecture. *Services Computing*, 222-229.

WIPO - World Intellectual Property Organisation (2013). International Patent Classification (IPC). http://www.wipo.int/classifications/ipc/en/. Retrieved 29/10/2013.

Chih-Chun Chen

Dr Chih-Chun Chen is a Post-Doctoral Research Associate at the Engineering Design Centre, University of Cambridge. She has a multi-disciplinary background with a PhD in Computer Science (University College London) and first degree in Psychology with Philosophy (University of Oxford). Her research interests lie mainly in the design and analysis of complex systems across different disciplines, both theory and practice. She has worked on a wide range of research projects involving a broad range of methods, including computational modelling of biological and social systems, statistical and network analyses of online communities and content, and empirical studies involving surveys and interviews.

Nathan Crilly

Dr Nathan Crilly is a Lecturer in Engineering Design at the University of Cambridge. His research interests are in the areas of design, technology and communication. He employs an interdisciplinary approach to studying how artefacts (e.g. products, systems or services) are developed, the properties they exhibit and the ways in which people respond to them. Nathan is a Fellow in Engineering at Clare College, Cambridge. He is a member of the Design Research Society and The Design Society. He also serves on the International Editorial Board of *Design Studies*.