SPECIFICATIONS FOR INNOVATION

Daniel P. Jenkins

DCA Design International, UK

Design specifications are a critical part of most development processes. Well constructed specifications describe constraints, focusing the development process and providing efficiency. However, by their very nature, they can constrain creativity by encouraging, and even forcing, designers to adopt current solutions rather innovating to meet the overall need. The way a development process is constrained is of key importance. Too few constraints, and efficiency and direction may be lost. Conversely, if there are too many constraints, opportunities for innovation may be missed. Using the example of a network connected thermostat, an approach is discussed that supports innovative design by describing, modelling and managing system constraints.

Introduction

Regardless if you are designing a train or a toothbrush, specifications can be used to provide explicit requirements. This often come in the form of physical dimensions (e.g. maximum size, location and size of interfacing parts), environmental factors (e.g. temperature and humidity to withstand), ergonomic factors (e.g. description of the target population), aesthetic or sensory factors (e.g. requirement to represent brand values), cost (e.g. material, manufacture, purchase), maintenance that will be needed, quality, and safety.

In regulated industries, such as transportation or medical devices, regulations and widely adopted standards often form the cornerstone of a design specification. Taking the example of door controls within trains, they prescribe acceptable locations (e.g. heights above floor level), arrangements (open above close) and actuation forces, along with specifications for reliability and robustness.

Detailed specifications are something that most engineers are very comfortable with, particularly in the later stages of the design process. When described appropriately, they provide measurable and testable requirements for an artefact – creating a clear description of what the product is, and how it should perform. Conversely, the prescriptive nature of a specification can also be viewed as stifling, particularly by designers in the early concept generation stages of a project. Tightly defined physical features and functions can limit the scope for lateral thinking. As such, it could reasonably be argued that design processes that are reliant on a product specification are better placed for evolutionary, as opposed to truly innovative products.

One approach used to mitigate the constraining nature of specifications is to discount, or dramatically restrict, the specification at the initial stages of the design process – delaying its introduction until after a series of concepts has been created. The classic example of this would be a ‘concept car’ that explores a new design direction without being overly concerned by details such as the construction techniques required, the material costs, or its impact on fuel performance. While this approach of ignoring constraints in the early stages of the process can have clear advantages for creativity, it can reinforce an ‘over the wall culture’ between design and engineering, playing up to role stereotypes. Furthermore, it could also be argued that an examination of constraints can direct creativity to develop new ways of managing these constraints.

Few would argue with the notion that detailed specifications can stifle creativity; however, their value to the design process, in terms of efficiency and focus, is also clearly evident. The natural question then, becomes; how can the system constraints be managed to ensure that the process retains the advantages of a prescriptive specification, without constraining the process of innovation? In short, as this paper will explain, it is contended that the consideration of constraints and a specification can, and should, exist throughout the design process. However, the level of detail, and means of presentation, should change to meet the requirement of the design activity at hand.

The role of specifications in teams

The importance of a design specification is particularly evident in projects with sizeable development teams and supply chains. A clear and auditable development process is of paramount importance and the design specification plays a critical role in this. ‘Ownership' for specific requirements can be assigned to individuals, regardless of whether they are in the core project team or cascaded to the supply chain. Likewise, where staged development processes are adopted, structured test plans can be employed to ensure that the design is compliant before project gateways are passed.

The reductional nature of a design specification is one of its great strengths in allowing the roles and responsibilities to be shared. However, without some form of systemic oversight, there is the very real danger that components, or constituent parts, may be designed to be compliant to the identified sections of the design specification, however, they may fail to adequately meet the purpose or values of the system. This is particularly relevant in cases where purposes are not cascaded with the requirements.

Specification generation

Where large organisations are ostensibly producing variants of the same core product (e.g. automotive companies, white goods producers), a highly structured process is particularly valued. The development process can remain common and be honed and refined based on previous practical experience (it is no coincidence that process improvement techniques such as 'six-sigma' have been widely adopted in these industries). At the start of the project, a template can be used to form the base specification. Input from different roles in the company such as marketing, ergonomics and benchmarking teams can be used to set specific values within the specification. These can describe the target audience, along with their requirements. The resulting high-level of consistency between these specifications has clear advantages for working with ambitious time scales. The specification provides focus and direction across the wider team and substantially reduces the duplication of effort and excessive exploration. From a management perspective, this also has a number of advantages. According to Klein (2014), organisations value predictability because they like projects to run smoothly. Companies like to plan the steps that will take each project from start to finish, the resources for each step, and the schedule. That way, managers can quickly notice perturbations and make the necessary adjustments. The converse to this is that where time pressures are critical, it can be far quicker and less risky to develop a variant of a proven product than strive for true innovation.

Developing better specifications

At the risk of oversimplifying things, the design process can be captured by the relationship between the following three words.

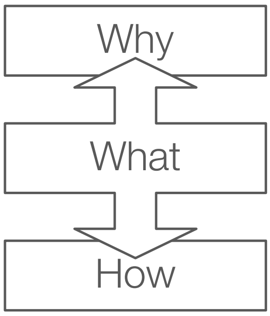

Figure 1: why-what-how triad

The specification should sit central to this, it describes in detail exactly *what* the artefact, that is being developed, should do. As the development process progresses, the design team adds detail to the design to explain *how* each of these requirements can be met. What can be missing in this process is the explicit link up to *why* the requirement or even the product exists.

Based on the description thus far, there are a number of shortfalls with the classic design process that is designed around a specification. Many design specifications could be improved by:

1. Creating an explicit link that describes *why* a requirement is needed
2. Allowing specifications to be viewed at differing levels of abstraction (i.e. what should it achieve at a physical level, what impact should it have on the end users life)
3. Describing the inter-relationships between components in a system that influence how requirements are to be met.

One means of addressing these challenges is to adopt a systems representation called the abstraction hierarchy (Rasmussen, 1986; see Figure 2). As the title suggests, the technique and resulting diagram describes a system at a number of levels of abstraction (typically five). At each level, the model can be used to describe what the system should do at a different level of abstraction. Moreover, each of the nodes in the model can be used to explore the 'what-why-how' triad introduced in Figure 1. A given node answers the question of what is required, while the linked nodes below describe how the system can achieve this, whereas the linked nodes above can be used to answer the question why. These upwards links are particularly valuable as they provide a rationale for the system. Where there are multiple connections from a single node, the diagram also describes the interrelationships between components and how the same affordances can be achieved by different physical objects.

Example

The utility of the abstraction hierarchy is perhaps best explained with an example – in this case a network-connected thermostat. While the idea behind the internet of things (IoT) has been around for quite some time, it is recently receiving unprecedented levels of interest. Ostensibly at least, the IoT involves creating a connection with everyday house hold objects (e.g. heating systems, locks, doors and windows, fans, lights) to allow them to be controlled remotely or automatically in response to events (e.g. a time of day, a change in environmental conditions, a message from a human). The Nest learning thermostat is used by many as the de facto standard example when describing the IoT and connected devices. Due to its familiarity, it has been adopted here for the purpose of explaining the approach.

The physical functions of a product are described at the base of the abstraction hierarchy, while at the top, it purpose, or why it exists is captured (see Figure 2). The top row of the diagram, the domain purpose, describes the overall purpose of the product, in this case to reduce energy consumption while maximising the user experience. Unlike goals, these objectives do not change with time or as a result of different events, but remain fixed. The level below, domain values, provides high-order measures of performance used to determine whether or not the functional purposes are being achieved. In this case, to maximise thermal comfort and convenience, while minimising environmental impact and energy usage.

Moving to the very bottom of the hierarchy, the physical objects are described. Within the thermostat itself these include component such as temperature sensors, display screens, rotary dials, etc. In addition, other components in the wider system are also included such as smart phones and apps they may be required to interface with the product. Other artefacts that may have overlapping affordances can also be captured (such as smart meters and utility bills). The functions of each of these objects are described above in the physical functions row above. These physical functions are the affordances or functions of the physical objects (what they do), independently of the purpose of the system. For example rotary dials capture user inputs, as opposed to capturing requests to change temperature. Describing function in this more generic way encourages the analyst to consider how functions and physical object can be used in different ways.

The row in the centre of Figure 2, the domain functions, links the diagram together. These are the functions that need to take place to meet the purpose of the system. For example ‘switch off heating when home is unoccupied’ or ‘adjust temperature to compensate for humidity’.

Together the diagram creates an explicit link between the functions of the physical object at the base of the hierarchy and the user values and purposes at the top. The complexity of the linking provides an indication of how these different needs can be met by a combination of physical objects. The diagram has been coded to highlight what the NEST adds over and above a traditional heating system at different levels of abstraction. As such, the impact of the introduction of new components (physical objects) can be considered. For example, the introduction of humidity sensor allows ambient humidity to be measured, this allows the temperature to be adjusted to compensate for humidity which, in turn, maximises thermal comfort while minimising energy usage and environmental impact. This has the potential to positively impact both purposes of reducing energy consumption and maximising user experience. Explicitly considering these high-order impacts can help to reduce ‘function-creep’ where components are added as a result of technical ease rather than a user requirements.

At the base level of the hierarchy, each link can be taken in turn and questioned to establish if there are other components already in the systems that can perform the same functions. Similarly, other functions or affordances of the existing components can be considered to see if they have the ability to have a positive impact on the overall system goals. The boundaries of the system can also be expanded to consider other products and components and how functions can be shared between these. This is of particular interest in the IoT where numerous devices are to be connected.

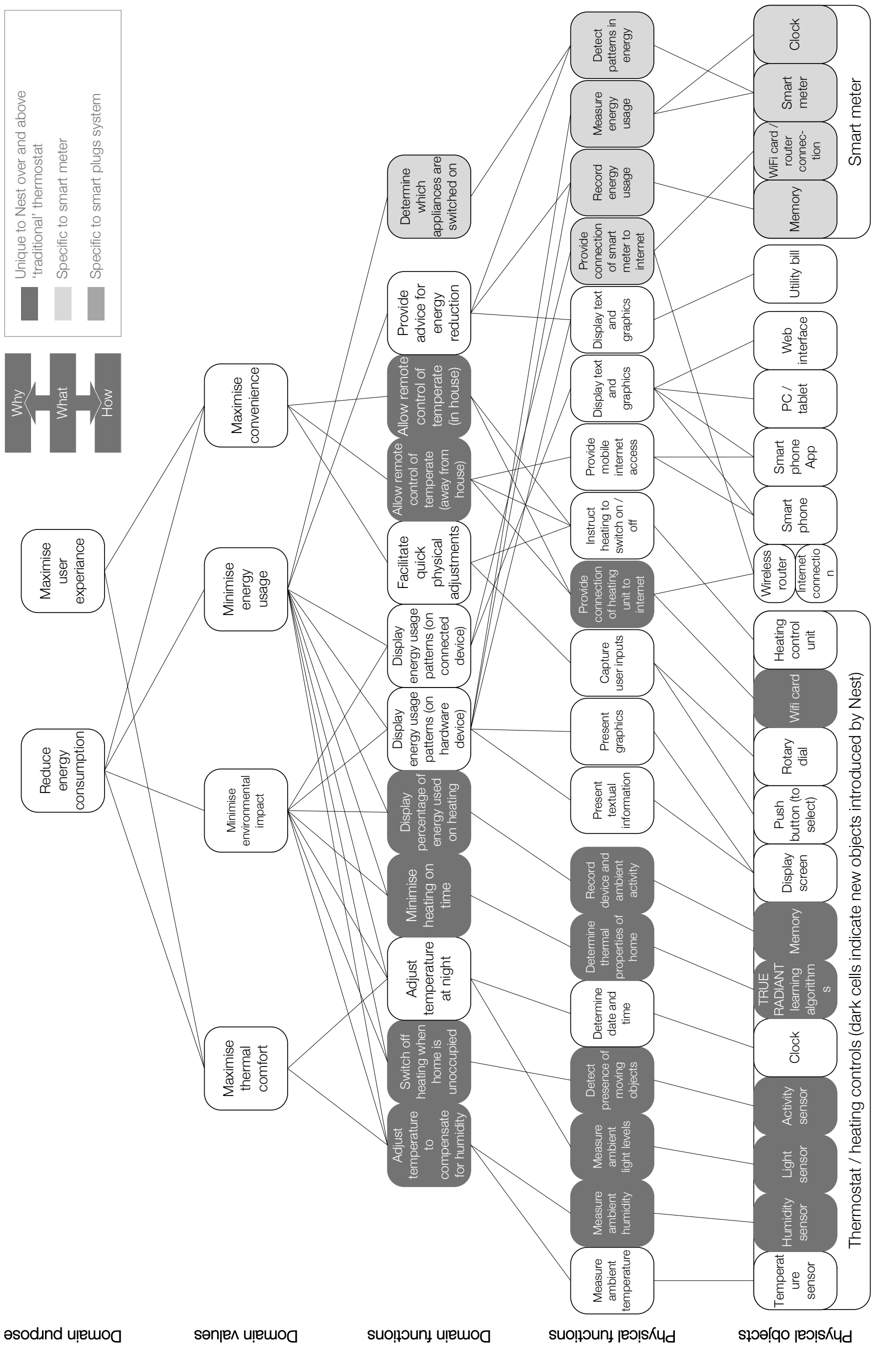

Figure 2: Abstraction hierarchy

The abstraction hierarchy provides a description of the system constraints at a high level. Furthermore, it encourages the analyst to consider the relationship between physical objects in the systems and the high-order purpose of the system. Accordingly, it lacks the detail of a traditional product specification. In most cases the design process would start with the abstraction hierarchy while the overall system architecture is being decided. The approach described is in no way intended to replace the standard specification, rather it aim to support it. Once a system architecture is agreed the abstraction hierarchy would inform the development of a more detailed design specification. This would include all of the traditional constraints such as relevant legislation, cost, size, safety, context restraints, and time. Explicit links can be made between these two representations (the abstraction hierarchy and the specification) allowing a clear audit trail and also allowing the specification to be updated in line with changes to the abstraction hierarchy should constraints or assumptions be modified.

Discussion

The challenge of generating a connected device can, of course, be viewed solely at a physical level by augmenting the current thermostat system with a means of communication. An existing detailed design specification from a legacy product could simply be modified to add a new requirement for connectivity. Accordingly, this connectivity can be viewed as a 'bolt-on' function – and the resultant focus would be on the decision of which type of communication protocol to adopt (e.g. WiFi or low energy radio communications) to reduce cost and increase reliability. Indeed an approach similar to this is likely to have been adopted by a number of manufacturers prior to the release of the Nest and deployed with varying level of success and adoption.

It is perhaps interesting though, that it was a small independent, albeit well-funded, company that designed and developed the product that is now viewed as the most innovative product. It was the Nest learning Thermostat, not a product from one of the large organisations that decided to move beyond a technical innovation, to one that seeks to actively engage the end user to meet the higher-order domain values. A number of the incumbent companies subsequently produced competitors to the Nest, offering products with similar components and functions at a physical level. However, by this point, benchmarking the competition would have allowed them to develop a detailed specification of what these products should do and the individual requirements for each of the components.

It would, of course, be naïve to think that the reason this level of innovation came from a start-up, and not a large organisation, was solely down to large organisations focusing on the physical and functional level of a project – brought about by over-reliance on design specifications. However, what is clear is that such a reliance on a 'traditional' product development process is not embracing opportunities for innovation.

The abstraction hierarchy is, again of course, not unique – there are a wide range of tools and techniques that encourage the design team to focus of user and stakeholder needs and seek to find innovative solutions to identified markets and problems (it is not clear what approach was used to develop the Nest product). Likewise, the approach described in the paper is certainly not proposed as a silver-bullet allowing a perfect balance to be struck between project efficiency and innovation. However, the described approach is proposed to help inform these tradeoffs, to encourage debate early on in the project of the purpose of the system.

References

Klein. G. (2014) No, your organisation really won't innovate'. Wired. Accessed 05/09/2014 http://www.wired.co.uk/magazine/archive/2014/05/ideas-bank/gary-klein

Rasmussen, J. (1986). Information processing and human-machine interaction: An approach to cognitive engineering. Amsterdam, The Netherlands: North-Holland.