Commonalities across Designing: Evidence from Models of Designing and Experiments

John S Gero and Udo Kannengiesser

This paper presents evidence of commonalities across designing, derived from analyses of models of designing and from empirical data. The models of designing are part of the key literature in the domains of engineering design, software design and service design. The empirical data is based on thirteen highly heterogeneous case studies that differ in the designers' geographical location, expertise and discipline, in the specific design task, in the size and composition of the design team, and in the length of the design session. The method we used for analysing the models and the empirical data is independent of any of these domain- or situation-specific parameters. The results indicate that there are commonalities both across different models of designing and across different design protocols. These commonalities are related to the first occurrence of design issues in the design process, and to the continuity and the rate with which design issues are generated. Our findings provide preliminary support for the claim that designing can be studied as a distinct human activity that appears in different expressions but shares the same fundamental characteristics.

1. Introduction

Designing is a complex activity that has attracted a significant amount of attention from different research domains, trying to demystify its manifold process. One of the biggest challenges in this regard is to define designing as a unique activity while it is used in a vast range of domains such as engineering, software, graphical interfaces, and electronics, to name a few. Understanding the commonalities amongst different expressions of designing is a prime step in developing a universal understanding of design (Asimow 1962; Lawson 1980; Dym 1994).

Most previous research efforts can be classified as either theory-driven or empirical or data-driven. Theory-driven research focuses on comparing different models of designing. For example, in 1998 an international workshop organized by Grabowski et al. (1998) brought together design theorists from different disciplines, aiming to build a unified or universal design theory. However, one of the biggest issues in comparing different models of designing continues to be the diversity of terms used across disciplines that practice design and even within the same design discipline (Vermaas 2009).

Empirical research is concerned with studying the behaviour of designers when designing for different requirements, commonly using interviews and protocol studies. Despite their validity in obtaining insight into the thoughts of designers (Ericsson and Simon 1993, Cross et al. 1996), the *ad-hoc* dependency of these methods on the data has been a barrier for generalising the results of these studies across different designers, design situations and design researchers (Gero 2010). In addition, the complexity of designing *per se* makes aggregating the results of empirical studies in large, statistically significant scales a challenging task.

In this paper we use an approach to analysing multiple models of designing and multiple design protocols that allows studying them independently of any domain-specific or environmental parameters. It is based on the cumulative occurrence of design issues over the course of designing, coded according to the Function-Behaviour-Structure (FBS) design issue system. The results of applying this method indicate that there are significant commonalities across different models of designing and across different empirical datasets.

2. Source Data

2.1 Three Models of Designing

We chose three models of designing from separate design disciplines as a basis for our analyses: engineering design, software design, and service design.

Engineering design is a design discipline with a long tradition in developing models of designing. One of the most detailed and established models in this discipline is Pahl and Beitz' (2007) *Systematic Approach*, which was first published in its German edition in 1977. It describes designing as a sequence of four phases: (1) Task Clarification, (2) Conceptual Design, (3) Embodiment Design, and (4) Detail Design. Task Clarification is concerned with collecting, formulating and documenting the requirements of the product to be designed. Conceptual Design aims to identify the basic principles and outline of a design solution (or concept). Embodiment Design then elaborates the design into a layout that satisfies various technical and economic criteria. Detail Design finalises the design and prepares production documents. Each of the four phases comprises a set of activities that can involve iterations. Table 1 shows the details of Pahl and Beitz' Systematic Approach.

Table 1 Pahl and Beitz' (2007) Systematic Approach

Phases	**Activities**
1. Task Clarification	1.1 Define basic market demands 1.2 Define attractiveness demands of the market segment 1.3 Document customer-specific technical performance requirements 1.4 Refine and extend the requirements using the checklist and scenario planning 1.5 Determine demands and wishes
2. Conceptual Design	2.1 Abstract to identify the essential problems 2.2 Establish function structures: overall function – subfunctions 2.3 Search for working principles that fulfil the subfunctions 2.4 Combine working principles into working structures 2.5 Select suitable combinations 2.6 Firm up into principle solution variants 2.7 Evaluate variants against technical and economic criteria
3. Embodiment Design	3.1 Identify embodiment-determining requirements 3.2 Produce scale drawings of spatial constraints 3.3 Identify embodiment-determining main function carriers 3.4 Develop preliminary layouts and form designs for the embodiment-determining main function carriers 3.5 Select suitable preliminary layouts 3.6 Develop preliminary layouts and form designs for the remaining main function carriers 3.7 Search for solutions to auxiliary functions 3.8 Develop detailed layouts and form designs for the main function carriers ensuring compatibility with the auxiliary function carriers 3.9 Develop detailed layouts and form designs for the auxiliary function carriers and complete the overall layouts 3.10 Evaluate against technical and economic criteria 3.11 Optimise and complete form designs 3.12 Check for errors and disturbing factors 3.13 Prepare preliminary parts lists and production documents
4. Detail Design	4.1 Finalise details; complete detail drawings 4.2 Integrate into overall layout drawings, assembly drawings and parts lists 4.3 Complete production documents with production, assembly, transport and operating instructions 4.4 Check all documents for standards, completeness and correctness

The discipline of software design has also brought about several models of designing. Here, one of the most widely used models is the *Rational*

Unified Process (RUP). Although it was primarily developed as a commercial product, its basic concepts outlined by Kruchten (2004) form a publicly available and highly cited model of designing. RUP defines the following phases for software design processes: (1) Inception, (2) Elaboration, (3) Construction, and (4) Transition. Inception deals with understanding the requirements and defining the scope of the design. Elaboration specifies and prototypes the main features and architecture of the software design solution. Construction elaborates this solution by developing the complete set of features and implementing all the components of the software. Transition focuses on verifying design quality, manufacturing, and delivering the software to the user. Kruchten (2004) suggests this four-phase process to be executed iteratively. He also suggests that the specific activities within each phase are to be configured depending on the needs of the individual design project. On the other hand, he describes "typical iteration plans" (*ibid*, Chapter 16) that can be viewed as a representative sequence of activities that is likely to cover most instances of software design processes. Table 2 summarises the phases and activities in such a "typical" configuration of RUP.[1]

Table 2 Kruchten's (2004) Rational Unified Process

Phases	Activities
1. Inception	1.1 Analyze the problem 1.2 Understand stakeholder needs 1.3 Define the system 1.4 Manage the scope of the system 1.5 Refine the system definition
2. Elaboration	2.1 Decide which use cases and scenarios will drive the development of the architecture 2.2 Understand this driver in detail and inspect the results 2.3 Reconsider use cases and risks 2.4 Prototype the user interface 2.5 Find obvious classes, do initial subsystem partitioning, and look at use cases in detail 2.6 Refine and homogenize classes and identify architecturally significant ones; inspect results 2.7 Consider the low-level package partitioning 2.8 Adjust to the implementation environment, decide the design of

---

[1] For the Inception phase we use the workflow defined for the requirements discipline and omit the design project management activities that are included in Kruchten's "typical" Inception phase. We view these management activities as beyond the scope of a model of designing. For the Transition phase, where there are no "typical" activities defined, we use Kruchten's deployment workflow.

	the key scenarios, and define formal class interfaces; inspect results 2.9 Consider concurrency and distribution of the architecture 2.10 Inspect the architectural design 2.11 Consider the physical packaging of the architecture 2.12 Plan the integration 2.13 Plan integration tests and system tests 2.14 Implement the classes and integrate 2.15 Integrate the implemented parts 2.16 Assess the executable architecture
3. Construction	3.1 Plan system-level integration 3.2 Plan and design system-level test 3.3 Refine use-case realizations 3.4 Plan and design integration tests at the subsystem and system levels 3.5 Develop code and test unit 3.6 Plan and implement unit test 3.7 Test unit within a subsystem 3.8 Integrate a subsystem 3.9 Test a subsystem 3.10 Release a subsystem 3.11 Integrate the system 3.12 Test integration 3.13 Test the system
4. Transition	4.1 Plan deployment 4.2 Develop support material 4.3 Produce deployment unit 4.4 Beta test product

Service design is a more recent discipline with few existing process models. One of them is *Design for Six Sigma* (DFSS), which has been used to describe both designing products and designing services (or processes). One of the many variants of DFSS that is specific to designing services is the ICOV (Identify-Conceptualize-Optimize-Validate) model presented by El-Haik and Roy (2005). We will refer to this model as DFSS-ICOV in this paper. It proposes the following phases: (1) Identify, (2) Conceptualize, (3) Optimize, and (4) Validate. The Identify phase collects and analyses the requirements for the service to be designed, by listening to both the “voice of the customer” and the “voice of the business”. The Conceptualize phase determines the technical requirements and basic components of the service. The Optimize phase aims to configure the service in a way to achieve the best possible performance. The Validate phase tests and refines the service and prepares its launch. At the end of every phase in DFSS-ICOV there is a review to decide whether to proceed to the next phase or whether to rework some decisions. Table 3 shows the phases and activities described in this model.

Table 3 El-Haik and Roy’s (2005) Identify-Conceptualize-Optimize-Validate model (Design for Six Sigma)

Phases	**Activities**
1. Identify	1.1 Idea creation 1.2 Voice of the customer and business
2. Conceptualize	2.1 Concept development 2.2 Preliminary design
3. Optimize	3.1 Design optimization
4. Validate	4.1 Verification 4.2 Launch readiness

While there are obvious domain-specific differences between the three models, we can already extract a first commonality: All three models use four sequential phases with similar goals, Table 4. As designing proceeds through the four phases, its focus ultimately shifts from the design problem (phase 1) to the design solution (phase 4), with two intermediate stages: One stage (phase 2) generates a list of general concepts that have the potential of being used as starting points for synthesis of variations (“concept structure”). The other stage (phase 3) turns these general concepts into specific solutions with respect to formulated goals, constraints or resources (“solution structure”). This general four-phase model is consistent with the widely held understanding of designing as a progression from the abstract to the concrete (Roozenburg and Cross 1991; Welch and Dixon 1994; Hubka and Eder 1996).

Table 4 Common goals of the individual phases in Pahl and Beitz’ Systematic Approach, Kruchten’s RUP, and El-Haik and Roy’s DFSS-ICOV

Phase	**Systematic Approach**	**RUP**	**DFSS-ICOV**	**Overall goal**
1	Task Clarification	Inception	Identify	Understanding & defining the design problem
2	Conceptual Design	Elaboration	Conceptualize	Generating a concept structure
3	Embodiment Design	Construction	Optimize	Generating a solution structure
4	Detail Design	Transition	Validate	Finalising & delivering the design solution

There are other models of designing that have different (usually higher) numbers of design phases. For example, the VDI-2221 (VDI 1985) model in engineering design has seven phases. The DMADV (Define-Measure-Analyze-Design-Verify) model and the IDDOV (Identify-Define-Design-Optimize-Validate) model are variants of DFSS that both have five phases. However, the seven phases of VDI-2221 can easily be collapsed into the four phases of Pahl and Beitz' (2004) Systematic Approach. Similarly, most activities defined within the five phases of DMADV and IDDOV can be mapped onto DFSS-ICOV's four phases.

Each of the three models of designing describes detailed sequences of activities within the four design phases. The models differ not only in the number of these activities (29 in the Systematic Approach, 35 in RUP, and 7 in DFSS-ICOV), but also in the terms and concepts they use to describe the output of every activity. For a more detailed analysis, we need to map the specific concepts used in the models onto a uniform, generic coding schema. One such schema is the FBS design issue schema that has previously been used for analysing design protocols (Gero and McNeill 1998; Kan and Gero 2005). It consists of six design issues: Requirements, Function, Expected Behaviour, Behaviour derived from Structure (or, shorthand, Structure Behaviour), Structure, and Description.

Requirements: includes all expressions of customer or market needs, demands, wishes and constraints that are explicitly provided to the designers at the outset of a design task. For example, requirement issues include "technical performance requirements [...] articulated by the customer" (Pahl and Beitz 2007, p. 150), "stakeholder requests" (Kruchten 2004, p. 166), and "customer needs and wants" (El-Haik and Roy 2005, p. 84).

Function: includes teleological representations that can cover any expression related to potential purposes of the artefact. These representations may be flow-based or state-based (Chittaro and Kumar 1998). Unlike requirement issues, function issues are not directly provided to the designer; they are generated by the designer based on interpretations of requirement issues. Function issues in the Systematic Approach include "the intended input/output relationship of a system" (Pahl and Beitz 2007, p. 31) and some examples of needs related to safety, aesthetics or economic properties. Function issues in RUP include the notion of a use case as a "sequence of actions a system performs that yields an observable result of value to a particular actor" (Kruchten 2004, p. 98), and some "nonfunctional requirements" that "deliver the desired quality to the end user" (*ibid*, p. 159). Function issues in DFSS-ICOV include "service and process functional requirements" that are derived from those requirements provided by the customer (El-Haik and Roy 2005, p. 87).

Expected Behaviour: includes attributes that describe the artefact's expected interaction with the environment. They can be used as guidance or assessment criteria for potential design solutions. Expected behaviour issues in the Systematic Approach include "physical effects" describing the "working principles" of the interactions between different parts of the design object (Pahl and Beitz 2007, p. 40), as well as "technical, economic and safety criteria" used for design evaluation (*ibid*, p. 193). Similarly, Expected behaviour issues in RUP are captured by the "design model" that "consists of a set of collaborations of model elements that provide the behaviour of the system" (Kruchten 2004, p. 177), and "measurable testing goals" (*ibid*, p. 253) that are often subsumed in "nonfunctional requirements". Expected behaviour issues in DFSS-ICOV include "CTSs (critical-to-satisfaction requirements, also known as big *Y*s)" (El-Haik and Roy 2005, p. 33) and some "functional requirements" such as the (expected) "service time" (*ibid*, p. 96).

Structure Behaviour: includes those attributes of the artefact that are measured, calculated or derived from observation of a specific design solution and its interaction with the environment. Instances of structure behaviour must be of the same type as instances of expected behaviour, so as to allow comparing and evaluating design solutions. As a result, structure behaviour issues cover the same notions in the three models of designing as outlined for expected behaviour issues.

Structure: includes the components of an artefact and their relationships. They can appear either as a "concept structure" or a "solution structure", which are the outputs of phases 2 and 3 in Table 1. The former includes Pahl and Beitz' (2007, p. 40) "working surfaces" and "working materials", Kruchten's (2004, p. 174) "classes and subsystems", and El-Haik and Roy's (2005, p. 6) "design parameters". The latter includes Pahl and Beitz' (2007, p. 227) "layout" and "form", Kruchten's (2004, p. 256) "code", and El-Haik and Roy's (2005, p. 7) "detail designs".

Description: includes any form of design-related representations produced by a designer, at any stage of the design process. The descriptions presented in the Systematic Approach include sketches, CAD models, requirements lists, physical prototypes, calculations, and other documentation produced by mechanical engineers. Descriptions in RUP include storyboards, UML models, code files, test plans and other representations produced by software designers. Descriptions in DFSS-ICOV include House of Quality diagrams, FMEA worksheets, process maps, and concept selection matrices, among many others.

Every activity described in the three models of designing is concerned with generating or transforming one or more design issues. A set of classes of transformations is provided by the situated FBS framework (Gero and Kannengiesser 2004). They include the generic processes of interpretation, reflection, focussing, action and derivation. They allow refining the activities described in the models of designing as sets of more detailed design steps. Take the first activity, "Define basic market demands", described within Pahl and Beitz' design phase of Task Clarification. This activity requires as input the interpretation of a "development order" or "product proposal" that contains the product's desired "functionality and performance", which in the FBS design issue system is a requirement issue. Next, "basic market demands", such as "suitable for tropical conditions" and "P > 20 kW" (Pahl and Beitz 2007, p. 147), are constructed by the designer as "implicit requirements, i.e. they are not articulated by the customer" (*ibid*, p. 150). We map these market demands onto function and expected behaviour issues, generated by the designer's reflection on previous experiences. All requirement issues, function issues and expected behaviour issues are then compiled in a "requirements list" and "Quality Function Deployment (QFD)" diagrams (*ibid*, p. 145) that represent description issues produced by the designer's actions. Table 5 summarises the mappings of these five design steps in a logical sequence.

Table 5 The steps involved in Pahl and Beitz' activity of "Define basic market demands" and their mappings onto the FBS design issue system and the situated FBS framework

Design step	**Pahl and Beitz' description**	**FBS design issue**	**Process in situated FBS framework**
1	Receive "development order" or "product proposal"	Requirement	Interpretation
2	Identify basic market demands	Function	Reflection
3		Expected Behaviour	Reflection
4	Produce QFD diagrams and requirements list	Description	Action
5			Action

We applied this method of coding to all three models of designing, resulting in three sequences of design steps each of which produces one of the six design issues. The Systematic Approach had 87 steps, RUP had 100 steps, and DFSS-ICOV had 41 steps.

2.2 Thirteen Design Protocols

Our source data consists of thirteen design protocols produced by various research groups that also segmented and coded these protocols according to the FBS design issue scheme. The protocols differ from one another in multiple ways producing a highly heterogeneous data source. Table 6 presents the state space covered by the thirteen protocols, in terms of seven independent variables and their ranges of values.

Table 6 The state space covered by the thirteen design protocols

Variable	Range of values
Source location of data	Australia, Singapore, Taiwan, UK, USA – seven states: CA, IL, MN, UT, VA
Design task	Designing of: • assistive window raising device • assistive door opening device • novel thermal ink pen • software system to simulate road traffic controls • art gallery • teaching device • future personal entertainment system • coffee maker • pedometer to encourage running • commercial website
Participants' expertise	Professional designers, Undergraduate students, High school students
Participants' knowledge domain	Architecture, Business, Electronics, Ergonomics, Industrial design, Interface design, Mechanical Engineering, Mechatronics, Psychology, Software, Web design
Team size	From 2 to 9 designers
Team composition	Homogeneous, Heterogeneous
Length of design protocol (in number of segments)	From 192 to 1,280 segments

As shown in Table 6, the protocols originate from four different continents and address a wide variety of design tasks. The participants include designers with different levels of expertise and with varying education and training in different disciplines. The team sizes vary from small teams of only two designers to larger teams of up to 9 designers. Some of the teams are homogeneous (consisting of designers with the same knowledge background), while others are heterogeneous (consisting of designers with different knowledge backgrounds). The lengths of the design sessions vary from 192 to 1,280 segments of the coded protocols.

Table 7 shows the specific characteristics of each of the thirteen design protocols.

Table 7. The thirteen design protocols

	VARIABLES						
Design Protocol	**Source Location of Data**	**Design Task**	**Participants' Expertise**	**Particpants' Knowledge Domain**	**Team Size**	**Team Composition**	**Length of Design Protocol [number of segments]**
P1	Virginia, USA	Designing an assistive window raising device; design method: unstructured	Undergraduate students	Mechanical engineering	2	Homogeneous	891
P2	Virginia, USA	Designing an assistive window raising device; design method: brainstorming	Undergraduate students	Mechanical engineering	2	Homogeneous	614
P3	Virginia, USA	Designing an assistive door opening device; design method: morphological analysis	Undergraduate students	Mechanical engineering	2	Homogeneous	500
P4	United Kingdom	Designing a novel thermal ink pen	Professional designers	Electronics, mechatronics ergonomics, business	7	Heterogeneous	1280
P5	California, USA	Designing a software system to simulate road traffic controls	Professional designers	Software	2	Homogeneous	596
P6	Sydney, Australia	Designing an art gallery	Professional designers	Architecture	2	Homogeneous	192
P7	Utah, USA	Designing an assistive window raising device	High school students	–	2	Homogeneous	426
P8	Illinois, USA	Designing a teaching device using protoyping	Undergraduate students	Mechanical engineering, psychology	3	Heterogeneous	328
P9	Illinois, USA	Designing a teaching device without using protoyping	Undergraduate students	Mechanical engineering, psychology	3	Heterogeneous	424
P10	Singapore	Designing a future personal entertainment system	Undergraduate students	Industrial design	2	Homogeneous	418
P11	Singapore	Designing a coffee maker	Undergraduate students	Industrial design	2	Homogeneous	782
P12	Taipei, Taiwan	Pedometer to encourage running	Undergraduate students	Industrial design	2	Homogeneous	304
P13	Minnesota, USA	Designing a commercial-level website	Professional designers	Interface design, web design, business analyst	9	Heterogeneous	289

3. Analysis Method

What our two datasets – the three coded models of designing and the thirteen coded design protocols – have in common is that they represent designing as sequences of steps, where every step leads to the occurrence of a design issue. (The steps in design protocols are called “segments”.) This allows using the same approach for analysing the two datasets, namely by calculating the cumulative occurrence of every design issue across all steps of designing. The cumulative occurrence (c) of design issue (x) at design step (n) is defined as $c = \sum_{i=1}^{n} x_i$ where (x_i) equals 1 if design step (i) is coded as (x) and 0 if design step (i) is not coded as (x). Plotting the results of this equation on a graph with the design steps (n) on the horizontal axis and the cumulative occurrence (c) on the vertical axis will visualise the occurrence of the design issues. Figure 1 shows a general representation of such a graph.

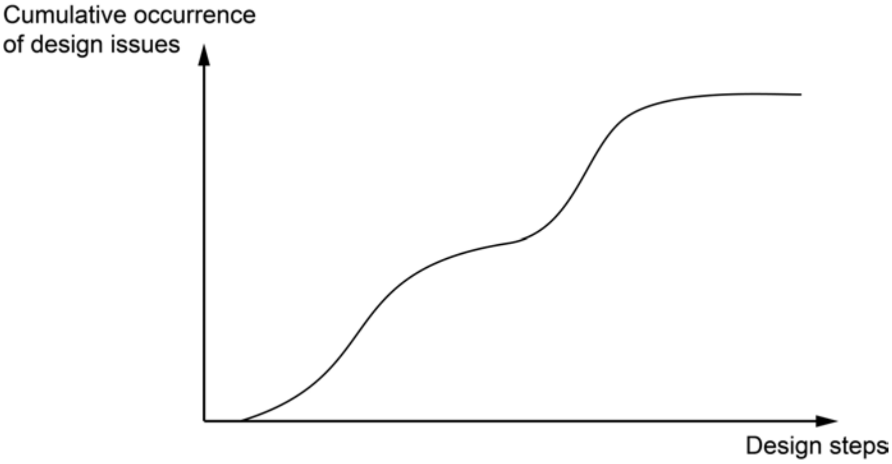

Fig1 Graphical representation of the cumulative occurrence of design issues across design steps

When using this representation for models of designing and design protocols, we need to keep in mind an important difference between the two. Models of designing are generic process models that compile any possible iteration or reformulation of the design into “loops”. These loops do not tell you how many times you have to execute them when carrying out a specific design process. Specific instances of design issues may occur multiple times when executing a loop; yet, the compiled process model accounts for them only as a single occurrence. In contrast, design protocols represent every instance of a design issue as a separate occurrence.

The following qualitative measures can be used for analysing models of designing and design protocols based on the cumulative occurrence of design issues:

- *First occurrence at start*: Which design issues first occur near the start of designing, and which first occur later?
- *Continuity*: Which design issues occur throughout designing, and which occur only up to a certain point?
- *Shape of the graph*: For which design issues is the cumulative occurrence graph linear, and for which is it non-linear?

In addition, we will use two quantitative measures:

- *Slope*: This is a measure for the speed at which design issues are generated.
- R^2 *(coefficient of determination)*: This is a measure for the linearity of the graph. We will set a minimum value of 0.950 as a condition for linearity.

All of these measures are independent of the number of design steps. This allows comparing models of designing that have different levels of detail. It also allows comparing design protocols that have different lengths in terms of their number of segments. Finally, we can compare models of designing against empirically-based design protocols.

4. Results

4.1 Cumulative Occurrence of Design Issues in Models of Designing

In this Section we present the qualitative and quantitative measures we derived from analysing the three models of designing. These measures are presented in Tables 8 to 13. In addition, to allow readers to carry out their own qualitative analyses, we also provide the raw data in the form of graphs representing the cumulative occurrence of design issues. These graphs are shown in Figures 2, 3 and 4. The vertical lines in these Figures separate the four phases in each model. They help in locating the occurrence of design issues within the respective model of designing, which is useful for deriving the two qualitative measures of “first occurrence at start” and “continuity”.

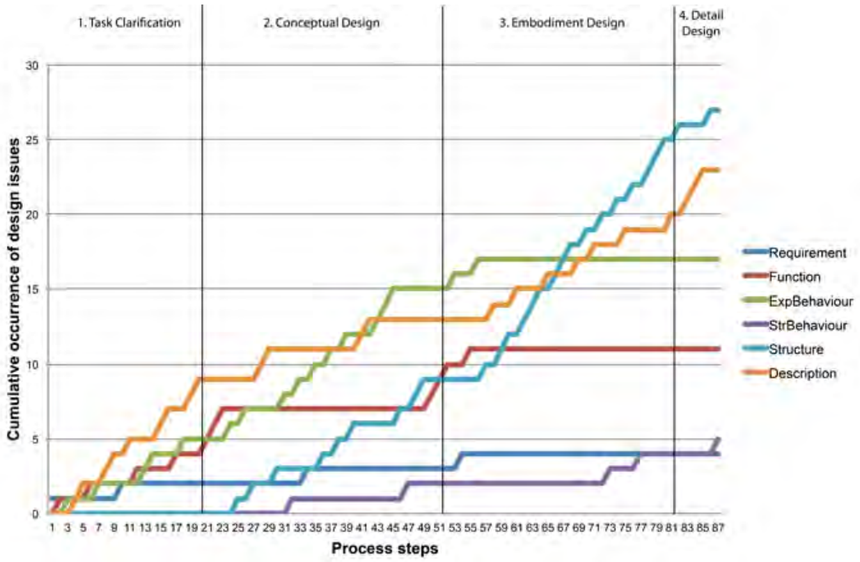

Fig2 Cumulative occurrence of design issues in the Systematic Approach

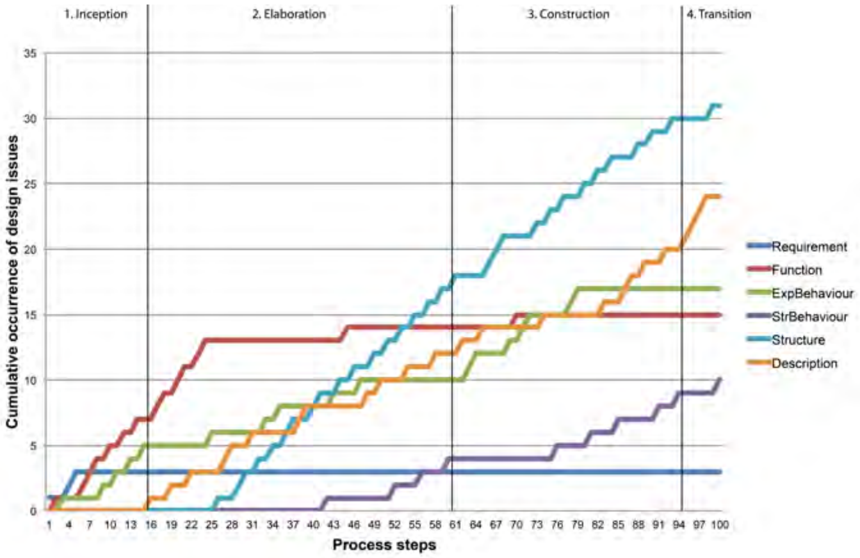

Fig3 Cumulative occurrence of design issues in the Rational Unified Process

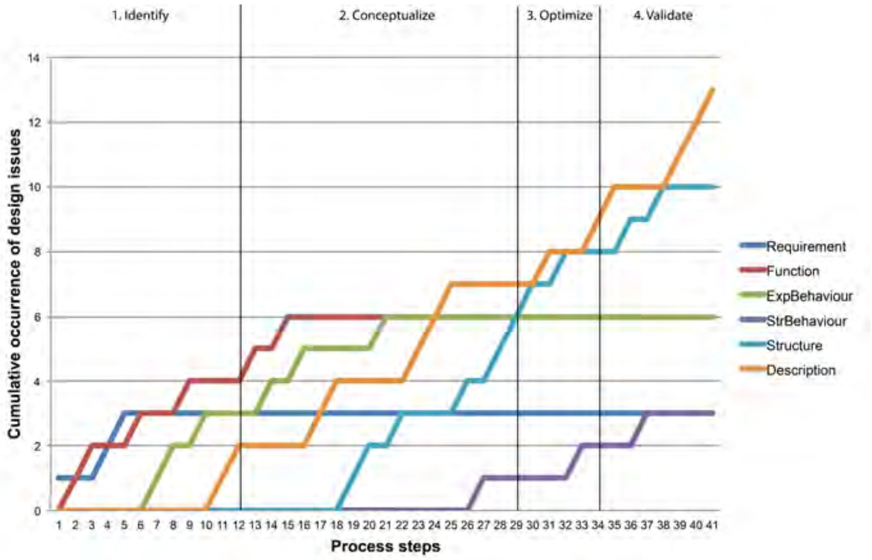

Fig4 Cumulative occurrence of design issues in DFSS-ICOV

Table 8 Quantitative and qualitative measures related to the cumulative occurrence of requirement issues in the three models of designing

Model of designing	Slope	R^2	First occurrence at start	Continuity	Shape
Systematic Approach*	---	---	Yes	No	---
RUP*	---	---	Yes	No	---
DFSS-ICOV*	---	---	Yes	No	---

* No statistical results produced due to small dataset (< 10 data points)

Table 9 Quantitative and qualitative measures related to the cumulative occurrence of function issues in the three models of designing

Model of designing	Slope	R^2	First occurrence at start	Continuity	Shape
Systematic Approach	0.127	0.889	Yes	No	Non-Linear
RUP	0.112	0.648	Yes	No	Non-Linear

DFSS-ICOV*	---	---	Yes	No	---
Mean	0.120	0.769			
Stdev	0.011	0.171			

* No statistical results produced due to small dataset (< 10 data points)

Table 10 Quantitative and qualitative measures related to the cumulative occurrence of expected behaviour issues in the three models of designing

Model of designing	Slope	R^2	First occurrence at start	Continuity	Shape
Systematic Approach	0.227	0.906	Yes	No	Non-Linear
RUP	0.179	0.972	Yes	No	Linear
DFSS-ICOV*	---	---	Yes	No	---
Mean	0.203	0.939			
Stdev	0.035	0.047			

* No statistical results produced due to small dataset (< 10 data points)

Table 11 Quantitative and qualitative measures related to the cumulative occurrence of structure behaviour issues in the three models of designing

Model of designing	Slope	R^2	First occurrence at start	Continuity	Shape
Systematic Approach*	---	---	No	Yes	---
RUP**	0.149	0.958	No	Yes	Linear
DFSS-ICOV*	---	---	No	Yes	---
Mean	0.100	0.920			
Stdev	0.069	0.053			

* No statistical results produced due to small dataset (< 10 data points)

** The first 40 design steps of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

Table 12 Quantitative and qualitative measures related to the cumulative occurrence of structure issues in the three models of designing

Model of designing	Slope	R^2	First occurrence at start	Continuity	Shape
Systematic Approach*	0.427	0.965	No	Yes	Linear
RUP	0.372	0.973	No	Yes	Linear
DFSS-ICOV**	0.448	0.977	No	Yes	Linear
Mean	0.416	0.972			
Stdev	0.039	0.006			

* The first 20 design steps of the protocol are ignored in slope and linearity calculation

** The first 15 design steps of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

Table 13 Quantitative and qualitative measures related to the cumulative occurrence of description issues in the three models of designing

Model of designing	Slope	R^2	First occurrence at start	Continuity	Shape
Systematic Approach	0.219	0.952	Yes	Yes	Linear
RUP	0.235	0.981	No	Yes	Linear
DFSS-ICOV	0.327	0.966	Yes	Yes	Linear
Mean	0.260	0.966			
Stdev	0.058	0.014			

Across the three models of designing, we can make the following observations regarding the three qualitative measures:

- First occurrence at start: In all three models, requirement issues, function issues and expected behaviour issues occur at the start (or in phase 1) of the design process. In two of the models (Systematic Approach and DFSS-ICOV), description issues also occur at the start. And in all three models, structure behaviour issues and structure issues occur later (in phase 2).
- Continuity: The cumulative occurrence of requirement issues, function issues and expected behaviour issues stagnates towards

  the end of designing (in phase 4) in all three models (i.e., it is discontinuous). Structure behaviour issues, structure issues and description issues are continuous in all three models.
- Shape of the graph: The cumulative occurrence of function issues in two of the models (Systematic Approach and RUP) is non-linear. The cumulative occurrence of structure issues and description issues in all three models is linear. For structure behaviour issues, only the analysis of RUP produces a result (linearity). For requirement issues the dataset in all three models is too small to make a meaningful statement, and for expected behaviour issues the results are inconsistent across the models.

4.2 Cumulative Occurrence of Design Issues in Design Protocols

Nine different sets of coders have been used by the different researchers to generate the empirical data. The coders arbitrated their coding using the Delphi method (Gero and McNeill 1998), discussing any differences until reaching agreement on the assigned codes. The average agreement between coders across the thirteen protocols is 89.8%.

Similar to the previous Section, we will present not only the results but also the raw data that is shown as sets of graphs representing the cumulative occurrence of design issues across all thirteen protocols. These graphs are not for measurement but for developing a qualitative understanding of the range and scale of the data. The graphs are of differing lengths, since each protocol has a different length.

The cumulative occurrences of requirement issues are shown graphically in Figure 5. Quantitative and qualitative measures are provided in Table 14 for all but six design protocols, which are indicated by the asterisks. In these six protocols the number of data points was too low (less than 10) to allow meaningful statements and statistical analyses. The remaining seven design protocols all exhibit the same qualities:

- Requirement issues in all protocols analysed occur from the start of the design session.
- Requirement issues in all protocols analysed occur discontinuously, as shown by the graphs tending to flatten out with increasing numbers of segments.
- The cumulative occurrence of requirement issues in all protocols analysed is non-linear. The mean R^2 value of 0.791 (standard deviation of 0.122) is below the threshold value of 0.950, set for linear modelling.

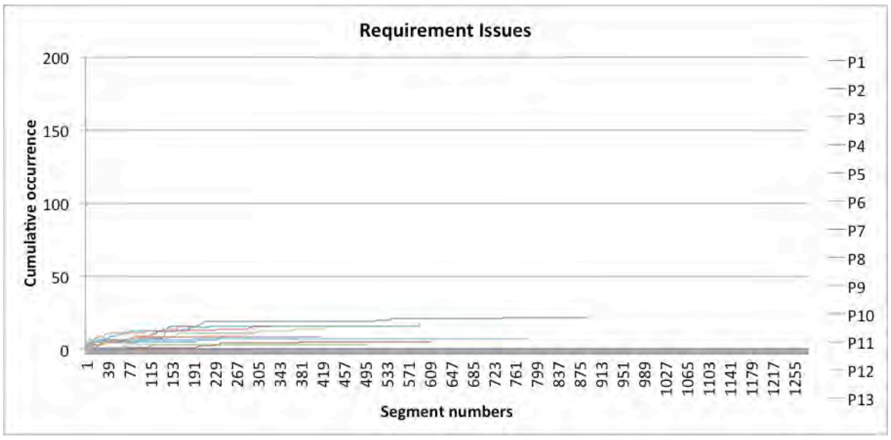

Fig5 Cumulative occurrence of requirement issues

Table 14 Quantitative and qualitative measures related to the cumulative occurrence of requirement issues in the thirteen design protocols

Protocol	Slope	R^2	First occurrence at start	Continuity	Shape
P1	0.018	0.646	Yes	No	Non-Linear
P2*	---	---	---	---	---
P3*	---	---	---	---	---
P4*	---	---	---	---	---
P5	0.014	0.621	Yes	No	Non-Linear
P6	0.055	0.791	Yes	No	Non-Linear
P7*	---	---	---	---	---
P8	0.043	0.882	Yes	No	Non-Linear
P9	0.028	0.900	Yes	No	Non-Linear
P10*	---	---	---	---	---
P11*	---	---	---	---	---
P12	0.025	0.772	Yes	No	Non-Linear
P13	0.047	0.928	Yes	No	Non-Linear
Mean	0.033	0.791			
Stdev	0.016	0.122			

* No statistical results produced due to small dataset (< 10 data points)

The cumulative occurrences of function issues are shown in Figure 6. The corresponding quantitative and qualitative measures are provided in Table 15, with the exception of five protocols that have too small datasets (as indicated by the asterisks in the table). Across the protocols we analysed, we can make the following observations:

- Function issues in all protocols analysed occur from the start of the design session.

- Function issues in most protocols analysed occur discontinuously, as their graphs flatten out towards the end of the design session. There are continuous occurrences in protocols P1 and P6; however, the total number of data points in these protocols (22 and 16, respectively) is fairly low, which makes their qualitative assessment less reliable.
- The cumulative occurrence of function issues in most protocols analysed is non-linear. The mean R^2 value is 0.888 (standard deviation of 0.071), which is below the threshold of 0.950. We found linearity only in protocol P3 (R^2 value of 0.960), yet based on a fairly small dataset (24 data points).

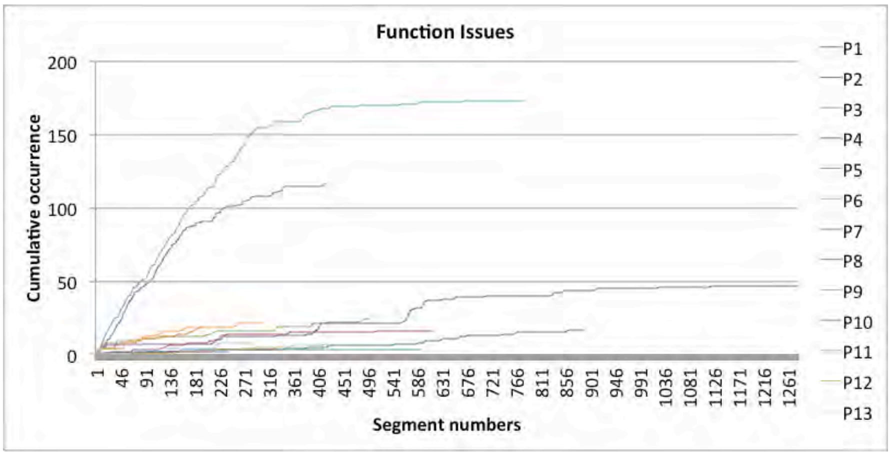

Fig6 Cumulative occurrence of function issues

Table 15 Quantitative and qualitative measures related to the cumulative occurrence of function issues in the thirteen design protocols

Protocol	Slope	R^2	First occurrence at start	Continuity	Shape
P1	0.019	0.929	Yes	Yes	Non-Linear
P2	0.028	0.830	Yes	No	Non-Linear
P3	0.034	0.960	Yes	No	Linear
P4	0.041	0.923	Yes	No	Non-Linear
P5*	---	---	---	---	---
P6	0.074	0.948	Yes	Yes	Non-Linear
P7*	---	---	---	---	---
P8*	---	---	---	---	---
P9*	---	---	---	---	---

P10	0.271	0.884	Yes	No	Non-Linear
P11	0.190	0.745	Yes	No	Non-Linear
P12	0.064	0.883	Yes	No	Non-Linear
P13*	---	---	---	---	---
Mean	0.090	0.888			
Stdev	0.091	0.071			

* No statistical results produced due to small dataset (< 10 data points)

The cumulative occurrences of expected behaviour issues are shown graphically in Figure 7. Quantitative and qualitative measures are summarised in Table 16.

Most of the thirteen protocols exhibit similarities:

- Expected behaviour issues in most protocols occur from the start of the design session. There is one exception in protocol P2; expected behaviour issues here occur with some delay.
- Expected behaviour issues in most protocols occur continuously. Exceptions are protocols P2, P3 and P11, where the occurrence of expected behaviour issues drops off towards the end of the design sessions.
- The cumulative occurrence of expected behaviour issues in most protocols is linear. The mean R^2 value is 0.972 (standard deviation of 0.022), which is above our threshold of 0.950. Only two protocols exhibit non-linearity in the occurrence of expected behaviour issues; these are protocols P3 and P11. This is probably related to the discontinuity observed in these protocols.

The mean slope of the expected behaviour issues graphs is 0.213, with a standard deviation of 0.135.

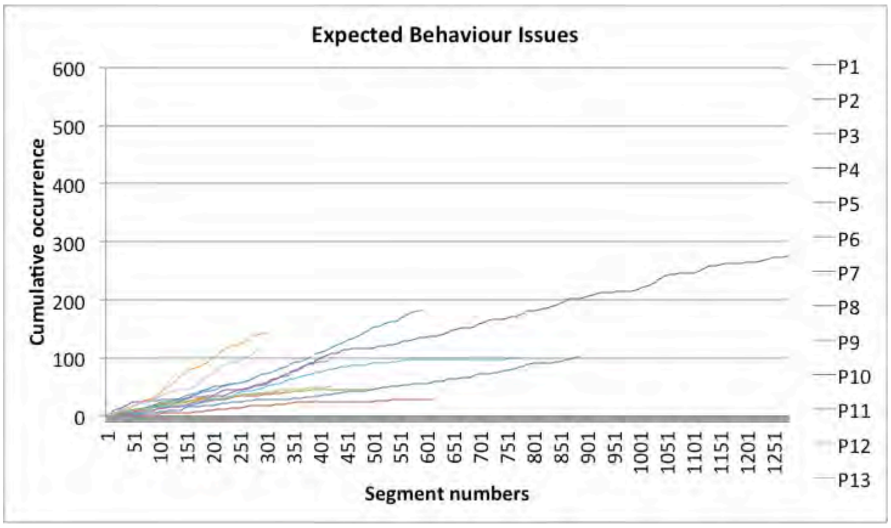

Fig7 Cumulative occurrence of expected behaviour issues

Table 16 Quantitative and qualitative measures related to the cumulative occurrence of expected behaviour issues in the thirteen design protocols

Protocol	Slope	R^2	First occurrence at start	Continuity	Shape
P1	0.110	0.984	Yes	Yes	Linear
P2	0.056	0.954	No	No	Linear
P3	0.090	0.929	Yes	No	Non-Linear
P4	0.222	0.995	Yes	Yes	Linear
P5	0.314	0.986	Yes	Yes	Linear
P6	0.175	0.975	Yes	Yes	Linear
P7	0.240	0.979	Yes	Yes	Linear
P8	0.130	0.989	Yes	Yes	Linear
P9	0.118	0.981	Yes	Yes	Linear
P10	0.239	0.959	Yes	Yes	Linear
P11	0.150	0.930	Yes	No	Non-Linear
P12	0.530	0.993	Yes	Yes	Linear
P13	0.397	0.984	Yes	Yes	Linear
Mean	0.213	0.972			
Stdev	0.135	0.022			

The cumulative occurrences of structure behaviour issues are shown in Figure 8, with Table 17 providing quantitative and qualitative measures, except for one protocol, P13 (as indicated by the asterisk in Table 17), that had too few occurrences of structure behaviour issues to be taken into account.

Again, there are strong similarities across the protocols analysed:

- Structure behaviour issues in most protocols occur from the start of the design session. There are exceptions in protocols P3, P10 and P11, where structure behaviour issues occur with some delay.
- Structure behaviour issues in all protocols analysed occur continuously.
- The cumulative occurrence of structure behaviour issues in most protocols is linear. The mean R^2 value is 0.982 (standard deviation of 0.019), which is above the threshold of 0.950. Only one protocol, P12, exhibits non-linearity.

The mean slope of the graphs is 0.246, with a standard deviation of 0.092.

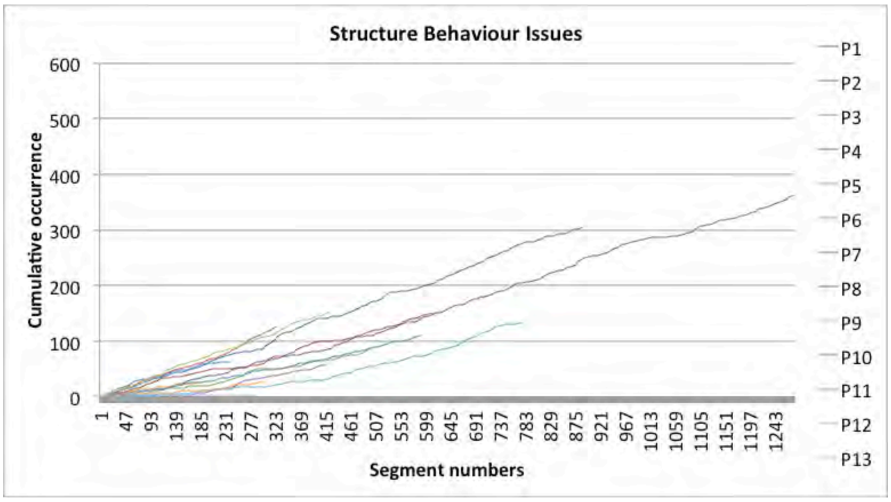

Fig8 Cumulative occurrence of structure behaviour issues

Table 17 Quantitative and qualitative measures related to the cumulative occurrence of structure behaviour issues in the thirteen design protocols

Protocol	Slope	R^2	First occurrence at start	Continuity	Shape
P1	0.352	0.997	Yes	Yes	Linear
P2	0.235	0.987	Yes	Yes	Linear
P3	0.179	0.982	No	Yes	Linear
P4	0.296	0.995	Yes	Yes	Linear

P5	0.186	0.991	Yes	Yes	Linear
P6	0.138	0.973	Yes	Yes	Linear
P7	0.283	0.975	Yes	Yes	Linear
P8	0.372	0.989	Yes	Yes	Linear
P9	0.361	0.998	Yes	Yes	Linear
P10	0.219	0.992	No	Yes	Linear
P11	0.254	0.974	No	Yes	Linear
P12	0.079	0.928	Yes	Yes	Non-Linear
P13*	---	---	---	---	---
Mean	0.246	0.982			
Stdev	0.092	0.019			

* No statistical results produced due to small dataset (< 10 data points)

The cumulative occurrences of structure issues are shown in Figure 9, with Table 18 providing quantitative and qualitative measures.

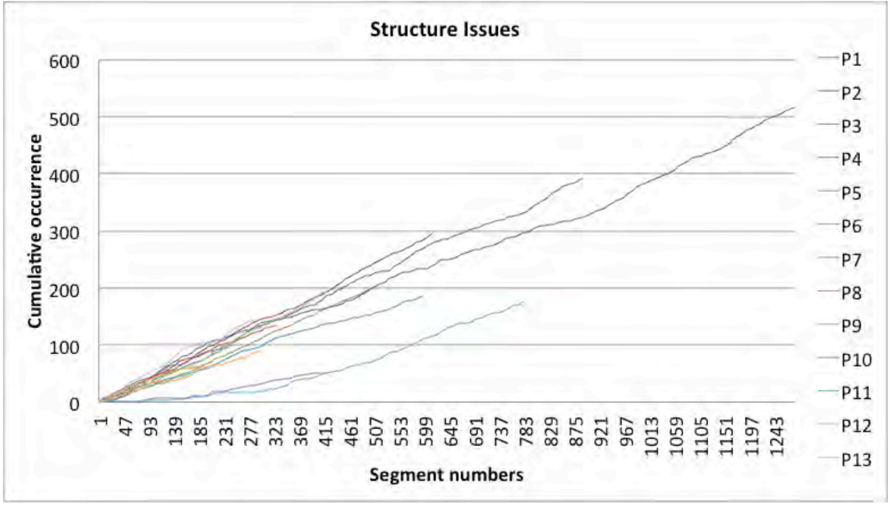

Fig9 Cumulative occurrence of structure issues

Table 18 Quantitative and qualitative measures related to the cumulative occurrence of structure issues in the thirteen design protocols

Protocol	Slope	R^2	First occurrence at start	Continuity	Shape
P1	0.437	0.999	Yes	Yes	Linear
P2	0.476	0.999	Yes	Yes	Linear
P3	0.417	0.998	Yes	Yes	Linear
P4	0.378	0.993	Yes	Yes	Linear
P5	0.313	0.994	Yes	Yes	Linear
P6	0.372	0.988	Yes	Yes	Linear

P7	0.411	0.997	Yes	Yes	Linear
P8	0.424	0.998	Yes	Yes	Linear
P9	0.469	0.995	Yes	Yes	Linear
P10*	0.186	0.993	No	Yes	Linear
P11**	0.336	0.993	No	Yes	Linear
P12	0.287	0.990	Yes	Yes	Linear
P13	0.507	0.989	Yes	Yes	Linear
Mean	0.386	0.994			
Stdev	0.088	0.004			

* The first 160 segments of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

** The first 300 segments of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

Commonalities across the thirteen protocols include:

- Structure issues in most protocols occur from the start of the design session. There are exceptions in protocols P10 and P11, where the designers did not generate structure issues until later in the design session.
- Structure issues in all protocols occur continuously.
- The cumulative occurrence of structure issues in all protocols is linear. The mean R^2 value is 0.994 (standard deviation of 0.004), which is above the threshold of 0.950. In this analysis we ignored the initial segments of P10 and P11 based on the late beginning of a clearly linear part of the graphs representing these protocols.

The mean slope of the graphs is 0.386, with a standard deviation of 0.088.

The cumulative occurrences of description issues are shown in Figure 10. Quantitative and qualitative measures are summarised in Table 19. Protocols P8, P9, P12 and P13 were not taken into account in this analysis because of the small dataset they provide for description issues.

We can observe the following commonalities:

- Description issues in most protocols do not occur from the start. Exceptions include protocols P2, P3 and P10.
- Description issues in most protocols occur continuously, except in P1 and P4.
- The cumulative occurrence of description issues in most protocols is linear, except in P4 and P7. The mean R^2 value is 0.964 (standard deviation of 0.036), which is above the threshold of 0.950. In this analysis we ignored the initial segments of P6 based on the late beginning of a clearly linear part of the graph representing this protocol.

The mean slope of the graphs is 0.166, with a standard deviation of 0.086.

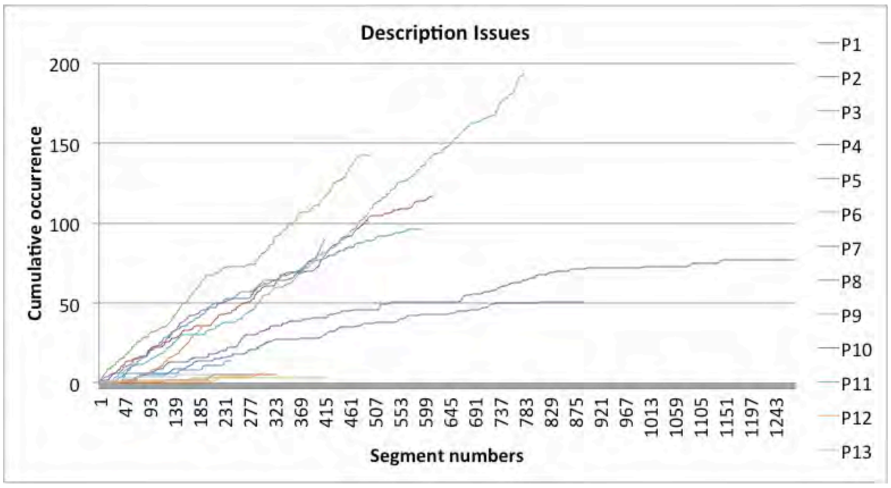

Fig10 Cumulative occurrence of description issues

Table 19 Quantitative and qualitative measures related to the cumulative occurrence of description issues in the thirteen design protocols

Protocol	Slope	R^2	First occurrence at start	Continuity	Shape
P1	0.064	0.970	No	No	Linear
P2	0.196	0.994	Yes	Yes	Linear
P3	0.274	0.992	Yes	Yes	Linear
P4	0.063	0.934	No	No	Non-Linear
P5	0.170	0.973	No	Yes	Linear
P6*	0.238	0.962	No	Yes	Linear
P7	0.051	0.881	No	Yes	Non-Linear
P8**	---	---	---	---	---
P9**	---	---	---	---	---
P10	0.192	0.979	Yes	Yes	Linear
P11	0.249	0.986	No	Yes	Linear
P12**	---	---	---	---	---
P13**	---	---	---	---	---
Mean	0.166	0.964			
Stdev	0.086	0.036			

* The first 46 segments of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

** No statistical results produced due to small dataset (< 10 data points)

5. Commonalities

5.1 Commonalities across Models of Designing

Our analysis has uncovered a number of commonalities among the three models of designing. We can characterise the support for these commonalities as:

- strong, if the analysis of all three models (i.e., three out of three models, 3/3) yields a common value for a specific measure
- weak, if the analysis of only two models (2/3 or 2/2) yields a common value for a specific measure
- not available, if there is no common value

Table 20 summarises the qualitative commonalities and their support across models of designing.

Table 20 Commonalities across models of designing. Their (proportional) support is shown in brackets.

Design issue	First occurrence at start	Continuity	Shape
Requirement	Yes (Strong)	No (Strong)	Not available
Function	Yes (Strong)	No (Strong)	Non-Linear (Weak)
Expected Behaviour	Yes (Strong)	No (Strong)	Not available
Structure Behaviour	No (Strong)	Yes (Strong)	Not available
Structure	No (Strong)	Yes (Strong)	Linear (Strong)
Description	Yes (Weak)	Yes (Strong)	Linear (Strong)

5.2 Commonalities across Design Protocols

The support for commonalities across the thirteen design protocols can be characterised in a similar way. Specifically, we define this support as:

- strong, if the analysis of at least 80% of the protocols yields a common value for a specific measure
- weak, if the analysis of at least 60% of the protocols yields a common value for a specific measure
- not available, if the analysis of less than 60% of the protocols

yields a common value for a specific measure

Table 21 shows the qualitative commonalities and their support across design protocols.

Table 21 Commonalities across design protocols. Their (proportional) support is shown in brackets.

Design issue	First occurrence at start	Continuity	Shape
Requirement	Yes (Strong)	No (Strong)	Non-Linear (Strong)
Function	Yes (Strong)	No (Weak)	Non-Linear (Strong)
Expected Behaviour	Yes (Strong)	Yes (Weak)	Linear (Strong)
Structure Behaviour	Yes (Weak)	Yes (Strong)	Linear (Strong)
Structure	Yes (Strong)	Yes (Strong)	Linear (Strong)
Description	No (Weak)	Yes (Weak)	Linear (Weak)

5.3 Comparison of Commonalities in Models of Designing and in Design Protocols

When comparing the commonalities across models of designing with the corresponding commonalities across design protocols, we can identify consistencies and inconsistencies. We define consistencies and inconsistencies as:

- significant, if the two commonalities are both strong
- noticeable, if one of the commonalities is weak
- slight, if the two commonalities are both weak

The consistencies and inconsistencies among models of designing and design protocols are shown in Tables 22 and 23, respectively.

Table 22 Consistencies among commonalities in models of designing and design protocols

Design issue	First occurrence at start	Continuity	Shape
Requirement	Yes (Significant)	No (Significant)	---
Function	Yes	No	Non-Linear

	(Significant)	(Noticeable)	(Noticeable)
Expected Behaviour	Yes (Significant)	---	---
Structure Behaviour	---	Yes (Significant)	---
Structure	---	Yes (Significant)	Linear (Significant)
Description	---	Yes (Noticeable)	Linear (Noticeable)

Table 23 Inconsistencies among commonalities in models of designing and design protocols

Design issue	First occurrence at start	Continuity	Shape
Requirement	---	---	---
Function	---	---	---
Expected Behaviour	---	No (models) vs Yes (protocols) (Noticeable)	---
Structure Behaviour	No (models) vs Yes (protocols) (Noticeable)	---	---
Structure	No (models) vs Yes (protocols) (Significant)	---	---
Description	Yes (models) vs No (protocols) (Slight)	---	---

In both the models and the protocols, requirement issues, function issues and expected behaviour issues occur from the start. However, there are inconsistencies related to the other design issues. Whereas structure issues and structure behaviour issues start occurring later in the models of designing, they start occurring right from the start in the design protocols. In turn, description issues start occurring from the start in the models, but later in the protocols. A conclusion we can draw from these comparisons is that even though designers engage in an initial phase of understanding the design problem, they simultaneously generate parts of a design solution. While this contradicts the models of designing, it confirms observations by some design researchers (Lawson 1980; Ullman et al. 1988) that designers tend to commit to specific solutions early on.

The models of designing and the protocols are largely consistent with respect to continuity. Requirement issues and function issues occur discontinuously; structure issues, structure behaviour issues and description issues occur continuously. The only inconsistency is related to expected behaviour issues: These issues occur discontinuously in the models and continuously in the protocols.

The shape of the graph in the models and the protocols is consistent for three design issues: a non-linear shape for function issues, and a linear shape for structure issues and description issues. Due to insufficient commonalities identified within the models of designing, we cannot make comparative claims related to the remaining three design issues.

6. Conclusion

Based on our analyses of models of designing and experimental data, we posit that there are regularities across designing that are independent of individual parameters. These parameters include the design discipline of the specific models, and the location, knowledge domain, expertise, team size, team composition, design task and length of the design session in the experiments. Many of these regularities can be seen as significant, based on the heterogeneity of the data and on the statistical evidence. It supports the premise that designing can be studied as a distinct human activity that transcends disciplinary boundaries and specific design situations (Cross 1982; Visser 2009).

The findings presented in this paper provide a starting point for two future research avenues. One avenue includes confirming and expanding our findings by examining more data. This involves analysing a wider range of models of designing and design protocols with additional parameters. Examples include studying design processes in collocated vs. remotely located teams, in single vs. multiple design sessions, using synchronous vs. asynchronous modes of communication, and with designers of different gender. The results may explain some of the exceptions or “outliers” we found in our analysis of design protocols, such as the designers’ delayed focus on structure issues in protocols P10 and P11. Other research may explore the significant phenomena in the commonalities we found, such as the strong focus of designers on structure issues, in terms of the high rate at which they are generated, and the high continuity and linearity with which they accumulate.

The other avenue includes investigating possible consequences of our findings for design education and design methodology. If we assume that designing is a distinct human activity that is independent of its many forms, then it may be useful to establish a common curriculum for designers, besides the current, discipline-specific study programs on mechanical engineering, industrial engineering, architecture, etc. The inconsistencies we found between models of designing and experiments may be used as focal points for expanding research into design methodologies. For example, the designers' early focus on structure issues that was observed in the experiments is the most significant discrepancy as to what current models of designing would predict. More scientific explanations are needed to explore this phenomenon and expand current models and theories of designing accordingly.

Acknowledgements

This research is supported in part by grants from the US National Science Foundation grant nos. SBE-0750853, EEC-0934824, CMMI–0926908 and IIS-1002079. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would like to thank the following for making their data available: Elizabeth Houle, University of Minnesota; Hao Jiang, National University of Singapore; Jeff Kan, Taylors University, Malaysia; Matthew Lammi, Utah State University; Marie Paretti, Virginia Tech; Hsien-Hui Tang, Taiwan University of Science and Technology; Christopher Williams, Virginia Tech; Robert Youmans, University of California at Northridge; Design Theory Research Symposium 7, London; and Studying Professional Software Designers Workshop, Irvine.

Parts of this paper were presented at the Fifth International Conference on Design Computing and Cognition 2012 (Gero et al. 2012).

References

Asimow M (1962) *Introduction to Design*, Prentice-Hall, Englewood Cliffs, NJ.

Chittaro L and Kumar AN (1998) Reasoning about function and its applications to engineering, *Artificial Intelligence in Engineering* **12**(4): 331-336.

Cross N (1982) Designerly ways of knowing, *Design Studies* **3**(4): 221-227.

Cross N, Christiaans H and Dorst K (1996) Analysing Design Activity. John Wiley & Sons, Chichester.

Dym C (1994) *Engineering Design: A Synthesis of Views*, Cambridge University Press, Cambridge, MA.

El-Haik B and Roy DM (2005) *Service Design for Six Sigma: A Roadmap for Ex-*

cellence, John Wiley & Sons, Hoboken, NJ.

Ericsson KA and Simon HA (1993) *Protocol Analysis: Verbal Reports as Data*. MIT Press, Cambridge, MA.

Hubka V and Eder WE (1996) *Design science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge*, Springer, London.

Lawson B (1980) *How Designers Think: The Design Process Demystified*, Architectural Press, Amsterdam.

Gero JS (1990) Design prototypes: A knowledge representation schema for design, *AI Magazine* **11**(4): 26-36.

Gero JS (2010) Generalizing design cognition research, *in* K Dorst et al. (eds) *DTRS8: Interpreting Design Thinking*, DAB documents, Sydney, pp. 187-198.

Gero JS and Kannengiesser U (2004) The situated function-behaviour-structure framework, *Design Studies* **25**(4): 373-391.

Gero JS, Kannengiesser U and Pourmohamadi M (2012) Commonalities across designing: Empirical results, *Design Computing and Cognition'12*, to appear

Gero JS and McNeill T (1998) An approach to the analysis of design protocols, *Design Studies* **19**(1): 21-61.

Grabowski H, Rude S and Grein G (1998) *Universal Design Theory*, Shaker Verlag, Aachen.

Kan J and Gero JS (2005) Design behaviour measurement by quantifying linkography in protocol studies of designing, *in* JS Gero and U Lindemann (eds) *Human Behaviour in Designing'05,* Key Centre of Design Computing and Cognition, University of Sydney, Australia, pp. 47-58.

Kruchten P (2004) *The Rational Unified Process: An Introduction*, Addison-Wesley, Upper Saddle River, NJ.

Pahl G and Beitz W (2007) *Engineering Design: A Systematic Approach*, Springer, Berlin.

Roozenburg NFM and Cross NG (1991) Models of the design process: Integrating across the disciplines, *Design Studies* **12**(4): 215-220.

Ullman DG, Dietterich TG and Stauffer LA (1988) A model of the mechanical design process based on empirical data, *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing* **2**(1): 33-52.

VDI (1985) *VDI-Richtlinie 2221 (Entwurf): Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*, VDI-Verlag, Düsseldorf.

Vermaas PE (2009) The flexible meaning of function in engineering, in M Norell Bergendahl et al. (eds) *Proceedings of the 17th International Conference on Engineering Design (ICED'09), Vol. 2*, The Design Society, pp. 113-124.

Visser W (2009) Design: one, but in different forms, *Design Studies* **30**(3): 187-223.

Welch RV and Dixon JR (1994) Guiding conceptual design through behavioral reasoning, *Research in Engineering Design* **6**(3): 169-188.