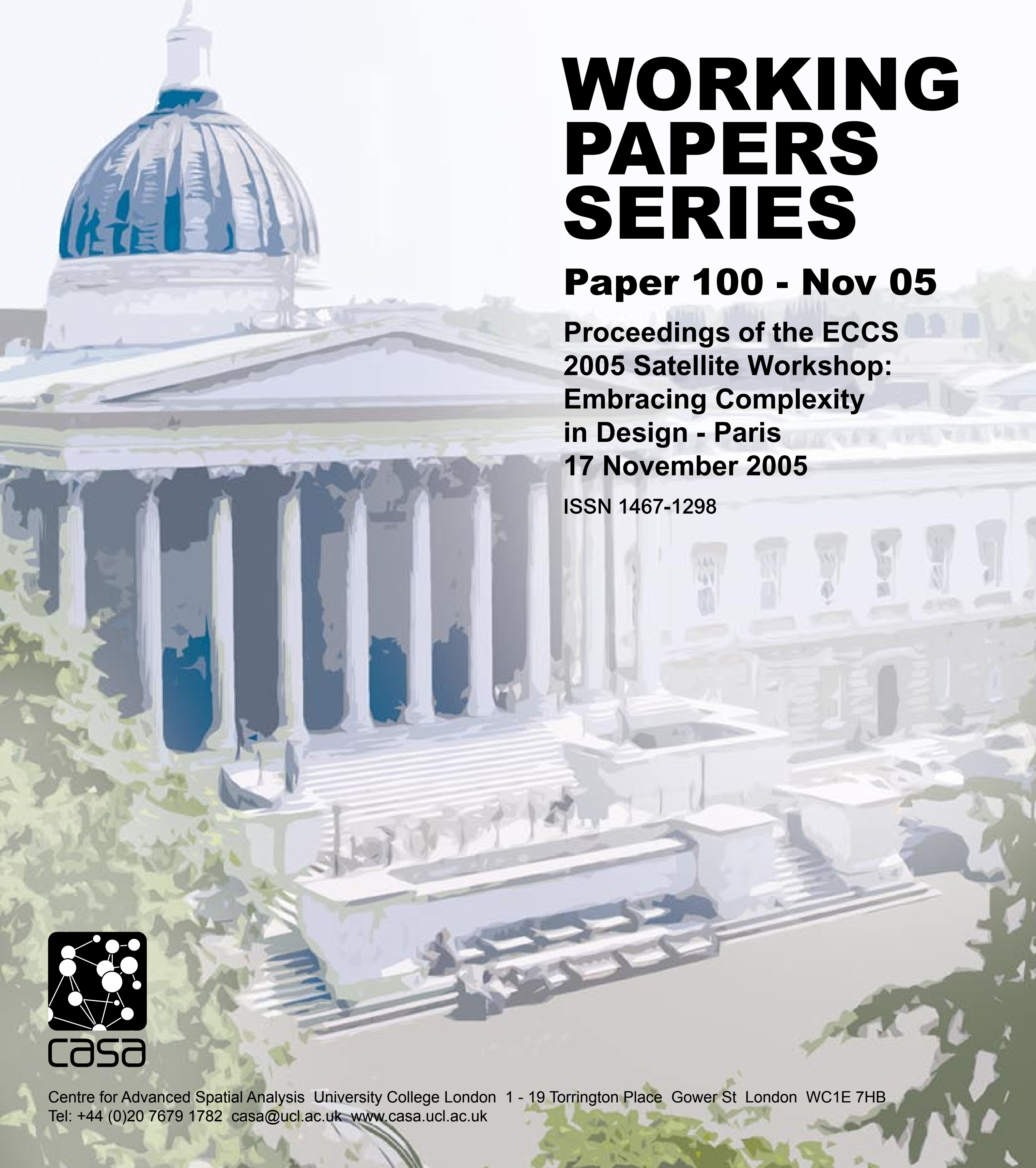
UCL CENTRE FOR ADVANCED SPATIAL ANALYSIS

WORKING PAPERS SERIES

Paper 100 - Nov 05

Proceedings of the ECCS 2005 Satellite Workshop: Embracing Complexity in Design - Paris 17 November 2005

ISSN 1467-1298



Centre for Advanced Spatial Analysis University College London 1 - 19 Torrington Place Gower St London WC1E 7HB
Tel: +44 (0)20 7679 1782 casa@ucl.ac.uk www.casa.ucl.ac.uk

Proceedings of the ECCS 2005 Satellite Workshop: **Embracing Complexity in Design**

Paris 17 November 2005

Edited by:

Jeffrey Johnson
Theodore Zamenopoulos
Katerina Alexiou

Published by The Open University

With support from the AHRC/EPSRC Designing for the 21st Century Research Cluster
Embracing Complexity in Design

EPSRC
Arts & Humanities Research Council
Information Society Technologies
CASA
The Open University

Proceedings of the ECCS 2005 Satellite Workshop: **Embracing Complexity in Design**

Paris 17 November 2005

Edited by:

Jeffrey Johnson
Theodore Zamenopoulos
Katerina Alexiou

Published by The Open University

With support from the AHRC/EPSRC Designing for the 21st Century Research Cluster
Embracing Complexity in Design

ISBN: 978-0-74921-545-3

Published by the Open University
Walton Hall
Milton Keynes
MK7 6AA
United Kingdom

© 2005 The Open University

Editorial

Embracing complexity in design is one of the critical issues and challenges of the 21st century. As the realization grows that design activities and artefacts display properties associated with complex adaptive systems, so grows the need to use complexity concepts and methods to understand these properties and inform the design of better artifacts. It is a great challenge because complexity science represents an epistemological and methodological swift that promises a holistic approach in the understanding and operational support of design. But design is also a major contributor in complexity research. Design science is concerned with problems that are fundamental in the sciences in general and complexity sciences in particular. For instance, design has been perceived and studied as a ubiquitous activity inherent in every human activity, as the art of generating hypotheses, as a type of experiment, or as a creative co-evolutionary process. Design science and its established approaches and practices can be a great source for advancement and innovation in complexity science.

These proceedings are the result of a workshop organized as part of the activities of a UK government AHRB/EPSRC funded research cluster called Embracing Complexity in Design (www.complexityanddesign.net) and the European Conference in Complex Systems (complexsystems.lri.fr). The purpose of the cluster is to create a research community and propose a research agenda on the relation between complexity and design. The hypothesis of the cluster is that complexity exists across every aspect of design, including:

- many designed *products and systems* are inherently complex, e.g. aeroplanes, buildings, cities, microchips, information systems, manufacturers, organisations.
- the social and economic *context* of design is complex, embracing market economics, legal regulation, social trends, mass culture, fashion, diffusion of innovation and much more.
- the *process* of designing can involve complex social dynamics, with many people processing and exchanging complex heterogeneous information over complex human and communication networks, in the context of many changing constraints
- designers need to understand the often complex dynamic processes used to fabricate and manufacture products and systems: *design, products and processes co-evolve*.

The aim of the workshop therefore, was to link together the design community and the complexity community and report the state of the art in research which exploits and encourages this cross-fertilization. The papers in these proceedings report theoretical, methodological and applied research on a variety of themes and suggest some interesting ways of bringing the linkage between complexity and design forwards.

The papers by Teymur, Jonas, Young, and Johnson investigate the relationships between complexity and design from their own perspectives. Teymur's paper accompanies an exhibition given at the workshop where the multiple definitions of the complexity of design are explored and presented in a visual and designerly fashion. Jonas is somewhat sceptical about the role complexity can play in design, writing that complexity sounds promising, but turns out to be a problematic and not really helpful concept. His paper moves away from "theories of what" towards practice and existing "theories of how" to design. He claims that using a systemic perspective leads to an evolutionary view allowing a clearer specification of the "knowledge gaps" inherent in the design process. Young proposes an integrated model of design with the aim of improving design practitioners' ability to navigate complex projects. Here the focus is on combining the description of different levels of design content with descriptions of the design process. Johnson approaches complexity and design from the perspective of complex systems, arguing that designers are the first scientists of artificial systems, and that the scientists of artificial systems are designers.

The paper by Thomson, Kumar, Chase and Duffy addresses the issue of measuring complexity within design projects, within design teams, and within Computer Aided Design environments. They suggest a generic framework for measuring complexity, which can be used in many design projects. In contrast, for Bittermann and Ciftcioglu complexity in design stems from time-varying design requirements. They argue that real time measurement of perceptual qualities of designs is necessary for an holistic approach in design, and propose a new method of geometric analysis termed Random Direction Distance Sampling which uses exponential averaging for time series.

The papers by den Besten and Dalle, and Cumming and Akar address different aspects of complexity issues in collaborative design. den Besten and Dalle are concerned with the relation between the complexity of design projects and the organisation of design teams. In particular, complex dynamics processes are involved in the design of open source software products. Their results tentatively suggest collaboration amongst developers is concentrated on those parts of the product characterised by complexity. Cumming and Akar introduce the concept of common ground, found within social linguistics and other domains, which concerns the contributions to mutual knowledge, mutual beliefs, and mutual assumptions that inform social and collaborative activity. Their paper then explores the implications of dynamically representing emergent common ground, and discusses how the representations can support collaborative design processes, giving details of how this can be implemented within Peer to Peer based design coordination applications.

The final paper in these proceedings by Zamenopoulos and Alexiou identifies some of the many ways in which design has embraced complexity by reviewing fundamental concepts and traditions developed within the two fields, and concludes with some suggestions of how design can inform complexity research.

These papers give a flavour of the wide ranging discussions our group has had over the last year, and the heterogeneous conclusions we have reached. We hope that you will find them inspirational and encourage you to join our community. In particular we hope that a wider community will discover that design theory and practice can contribute significantly to the science of complex systems.

Katerina Alexiou
Jeffrey Johnson
Theodore Zamenopoulos

The Open University

3rd November 2005

Table of Contents

Assessing the impact of product complexity on organizational design in open source software: Findings & future work.

Matthijs den Besten[a,b] and Jean-Michel Dalle[a,b]

[a] Université Pierre et Marie Curie, Paris, France – matthijs.denbesten@lamsade.dauphine.fr
[b] Université Paris-Dauphine, Paris, France – jean-michel.dalle@upmc.fr

ABSTRACT

We suggest an agenda for the investigation of the complex dynamic processes, notably teamwork, involved in the design and development of complex software products. We specifically investigate the division of labour among developers of open source software, by analyzing the logs of a set of open source software projects in order to monitor the evolution of development activity across contributors over time. Our first results suggest that collaboration among developers is concentrated on specific parts of the projects. What exactly distinguishes these parts from the rest of the project and attracts the developers is yet unclear, but we suggest that complexity at the level of the code could play a role.

Keywords: Open Source Software; Complexity; Virtual Teams

1. INTRODUCTION

Teams in general and virtual teams in particular enjoy an increasing interest from scholars in organizational science.[1,2] Characterized by the absence of a strong managerial hand, virtual teams do not make it obvious how team members collaborate – especially when the members are located in various parts of the world. Yet, in many circumstances virtual teams appear to be rather remarkably successful, but no clear understanding exists yet of the conditions of their success or efficiency. However, the work of virtual teams should be at least partly traceable in the activity logs that those teams leave behind in their virtual environments. This is the basic premise behind the study of developer activity in open source project logs that we present here, which is related to the work of other researchers who are also looking for ways to harness the wealth of data that emerges as a by-product of the project management and collaboration tools that these virtual teams employ.[3] Featuring free and easy access to such data together with increasing economic success, open source software projects are natural candidates for quantitative empirical studies of virtual teams.[4] Indeed, we would like to more specifically find out here whether it is possible to link the technical structure of the code within open-source software projects to the allocation of tasks between developers. Although the first results are encouraging, further research is needed, and we mostly aim here at setting out an agenda for future studies.

The next section introduces open source software and reviews some of the research done in that area. In section 3, we describe the data we studied and how we obtained them. It is followed, in section 4, by exemplifying the kind of research that the data allow us to do and by presenting in Section 5 an analysis of some preliminary results with respect to the allocation of tasks in virtual teams. We conclude by pointing out several avenues for further research.

2. OPEN SOURCE SOFTWARE

Open source software (OSS) is a type of software that has become increasingly prevalent over recent years. In contrast to closed source software, in OSS the human readable source code of the software program is distributed along with the program itself. With this source code it becomes then possible for users of the program to scrutinize the inner workings of the program and to adapt the program to their needs. The most famous example of OSS is Linux, an operating system developed based on Unix that is developed by Linus Torvalds and many other developers.[5] Microsoft, a dominant player in the market for operating systems, acknowledged the strength of Linux very early on, in what is now known as the "Halloween document"[6], and since then, the software industry has looked for ways to adapt features of the open source development model in more traditional closed environments.[7,8]

Yet, there is still something particular, and largely puzzling, about the OSS development model. In general, what is understood as the OSS development model is that it corresponds to the community-based voluntary self-organizing effort of various virtual teams of physically dispersed computer programmers to develop software – that is itself open to inspection to everyone who is interested. Eric Raymond famously likened the OSS development model to the interactions that are going on in a "bazaar".[9] However, since then, several case studies of open source software projects showed that in many projects hierarchies tend to persist and that there is larger diversity in organizational forms from one project to the other than would have been expected.[10] Indeed, in so far as there is a OSS development model, recent research seems to point towards an “onion model” of organization in which a core team of just a few developers is aided by a larger group of co-developers who are in turn aided by an even larger group of bug-submitters and feature-requesters, etc.[11] That is, open source development typically involves the participation of a large number of users who report bugs and request features, to be compared to a more limited number of co-developers who suggest software code that addresses those bugs and features; and to yet a smaller set of core developers who review the suggested code contributions and incorporate them in the existing code base.

What makes open source software projects particularly attractive as a topic for research is that virtually the whole development process is recorded and that the archives of these recordings are freely available for investigation. More in particular, open source software projects typically feature mailing lists where developers discuss their work and non-developers submit requests or ask for help. In addition, there may be discussion forums and bug tracking tools. Last, but not least, the source code is available and, when, as is often the case, a version control system is employed, in fact all old versions of the source code so that the development process can be traced back to the start. Researchers of software engineering have started to make use of this wealth of data to inform their investigations. Notable examples are the work of Walt Scacchi[12], who performed an in-depth ethnographical analysis of the implicit ways in which requirements are gathered in open source projects, and that of Mockus and Herbsleb[13], who studied the pace with which bugs were resolved based on information in mailing lists and software logs. Hashler and Koch[14] propose a larger scale mining of the available information and discuss what kind of questions could be explored on the basis of that information.

3. MINING DEVELOPMENT DATA

The data that we looked at for our particular investigation of the allocation of tasks in open source software project teams was extracted from logs of development activity that are maintained by software version control systems. Version control systems are used by development teams in order to keep track of what was contributed when by whom. If conflicts arise due to a change in the code, a version control system makes it possible to undo that change and revert to the source code as it was before the change was made. Note, however, that in most OSS projects, a possible change has already been thoroughly reviewed before it is allowed to be applied to the source code in the first place. Also, the people who commit the change are not necessarily the ones who wrote the code incorporated in that change. Rather, they are likely to be the *maintainers* of a part of the source code, who after a review of a change suggested by others, decide it is a good change and apply it to their part of the source code. In some cases, each change has to be approved of by a committee of core developers. In other cases, the review of suggested changes is completely up to the digression of the maintainer of the part of the source code to which the change is applied.

The logs that we looked at were logs kept by the CVS version control system. CVS is the most widely used version control system for open source software development and its logs are relatively easy to parse.[15] The procedure for obtaining the logs is rather straightforward: First, find the official website of the project you want to study. Then, check whether the project uses CVS and check whether the project allows outsiders to access the CVS repository. Most of the time this will be the case and there will be detailed instruction on how to obtain access to the CVS. Once you have obtained access and downloaded the current version of the source code, you can then issue the command “cvs log –N –b” and this will yield the log of the main branch of the development history. For now, we are only interested in the main branch. The log lists for each file each revision of that file and for each revision when the revision was made, who was responsible for the revision and how many lines of code were added to and deleted from the file as a result of the revision (see Annex). When a change applies to multiple files at the same time, it is registered for each of the files separately. If you are only interested in the number of times that a maintainer has been active and not in how many files he or she worked on, it is possible to merge the commits resulting from a change to multiple file back into one by comparing the commit dates of the revisions at the file level. Another thing that CVS allows you to do is to check out the actual code that is added or deleted during the revision. Finally, CVS make it possible to compare the whole of the source code at different points in time[*].

From the data in the CVS logs, we can gather various statistics. For one, we can find out how many people contributed code to the project, how often code was contributed, and to how many files. Besides, we can look at how the contribution behaviour changed over time. Finally, we can look at the distribution of development activity

---

[*] It should be kept in mind that the CVS data we study cover maintenance rather than programming activity since not all developers have CVS commit rights and since they therefore need to get their submissions through to developers with those rights, in one way or another. Only sometimes is it possible to track down who conceived the contributed code – when appropriate “credits” are given in the comments to the commit. However, even when everyone would be properly credited, we would not be able to tell which contributions were rejected or how much the code-author benefited from input from others.

among contributors over time. The information about the size of the change makes it possible to estimate the effort involved in each contribution and more precise estimates of the effort are possible by looking at the code of each contribution. More in particular, when looking at individual contributions, one could compute the Halstead complexity of the contribution or the change in McCabe complexity of the functions within the file that were affected by the contribution.[16] At the system level, one could identify a design-structure-matrix and determine the effect of contributions on the properties of that matrix.[17]

4. EXPLORING THE LOGS

We first selected a set of open-source projects. An attempt was made to obtain a set that was diverse in terms of product complexity, task uncertainty, and target audience. In addition, the projects needed to have a minimum amount of code, contributors and development history. Last, but not least, only those projects which provided easy access to their code repositories run by CVS, could qualify. In the end we settled for nine projects: An operating system – *NetBSD*, a data base – *PostgreSQL*, web server – *Apache*, a web browser – *Mozilla*, an instant message application – *Gaim*, a secure networking protocol – *OpenSSH*, a programming language – *Python*, a compiler – *GCC*, and a version control system – *CVS*. Several of these projects, most notably Mozilla and Apache, have already received a lot of attention from researchers. Others, like Gaim, stand out because of the amount of activity or because of the sheer length of activity.

However, it was not fully clear where the limits of a given project ended. For instance, Apache and Mozilla have their own repositories but both host multiple applications. Lacking a clear rule for now about where to draw these limits, we decided in favour of variety. In case of Apache, we restricted ourselves to the logs concerning Apache HTTP Server 2.0. In case of Mozilla, we considered the whole suite. In case of NetBSD, we only looked at the kernel of the operating system, while in the case of OpenSSH, which is part of OpenBSD, we focused at the subdirectory within OpenBSD where OpenSSH resides. Next, we extracted the CVS logs and computed the descriptive data displayed in Table 1. Similar data are available for a great number of projects elsewhere as well.[18] Note that the logs typically span a period of five to ten years. However, the earliest record in the log does not necessarily coincide with the start of the project itself as the decision to adopt CVS could have been made well into the development of the project: A case in point is GCC, which started well before the first recorded commit in 1997. Note also that in most projects the number of developers that are active in any given month is quite limited and that the total number of developers is usually much larger.

We then computed for each file that was logged and for each month how many distinct developers had committed a change to that file during that period. We found that in 80 to 90% of the cases, there had been only one developer. We then restricted our attention to files ending with .c, .C, .cc, or .cpp – suffixes indicating that the file in question contains code written in C or C++, two of the most widely used programming languages in software development, excluding in this way files that are not strictly part of the software source code. We still obtained very similar results. Consequently, we restricted our attention to active C and C++ files defined as those files to which at least 2 changes had been applied in the 30 days preceding the last recorded change[†]. For each of these, we computed a number of metrics and, employing a two-sided and then a one-sided t-test, we tried to determine for which of those metrics the files with only one committer during those last 30 days ("I-Mode") were different from those with multiple committers ("C-Mode"): I-Mode concerns developers working mainly in isolation and focusing on a specific open-source project, as they exist in quantity in repositories like SourceForge, while by contrast, C-Mode deals with most large open-source projects and involves a large number – a team – of interacting developers.[21] Results are shown in Table 2 (2-sided t-test) and 3 (one-sided t-test).

5. INTERPRETING THE RESULTS

At this level of investigation, we first note that projects seem to follow different development patterns – one further reason why inquiries about open-source projects should not focus on too limited a number of projects. We actually find very significant differences between C-mode and I-mode in some cases, and almost none in others. Further inquiries should try and correlate these patterns with macro-characteristics of projects, such as their age, or the total number of maintainers or even more specific maintenance policies. For instance, the lack of differentiation between C-mode and I-mode for PostgreSQL could be connected to its relatively limited number of maintainers (25: see Table 1). In a somewhat related manner, CVS and OpenSSH do not exhibit very clear patterns either, which might be explained by a relatively low activity going on in these smaller projects, or to too small a sub-sample of files to

---

[†] More precise rules, which we intend to study in later developments, would be to monitor activity on a longer period of time. Similarly, we might need to investigate whether CVS accounts could be used by more than one maintainer – another potential source of bias.

which we would have applied our statistical tests, or else to our definition of activity: on the contrary, it has probably nothing to do with the pooling of commits by maintainers since on average a lower number of SLOC is added (addM) on these 2 projects compared to others (see data in Annex).

Generally, patterns found tend to support the general distinction between I-Mode and C-Mode, and present preliminary evidence of the existence of these 2 different types of files *within the sample of actively maintained files*. C-Mode files appear generally significantly older, except for Apache, and tend to have received more contributions from more numerous maintainers. These results could suggest that these modules have "always been" in C-Mode, or conversely that C-Mode would be endogenous within large projects, that is, as we had hypothesized elsewhere[20], that maintainers and developers could be attracted by “hot spots” within which activity would be already going on. However, we cannot discriminate for now between these two hypotheses, and without more sophisticated statistical models. Indeed, similar results could hold if C-Mode files stayed active longer precisely because they had more than one maintainer.

Then, yet another potential explanation would have to do with the existence, or conversely the absence, of technical dependencies between the functions contained in these files, which would make the files either modules or simply libraries of functions. Indeed, C-Mode files also tend to have more numerous functions, and are generally bigger, except for Python and in a more limited measure for Gaim. This might either mean that C-Mode files grow by aggregating various functions and their respective maintainers, or that they are truly modules in the sense that they would be composed of numerous interacting functions. In that sense, C-Mode could cover various patterns of collective behaviours: Real teamwork in the context of modules composed of interacting functions, or “disguised” I-Mode where different maintainers would focus each on a different function in the context of a library of largely independent functions, which would have been grouped in the same files for tractability reasons. Here again, only a more developed model will help us to discriminate more precisely. Indeed, a more general difficulty in interpreting these results has to do with modularity and the extent of modularisation in open-source projects. That is, the technical design of software products does impose some collaborations as a function of the dependencies among the parts of the code that developers work on. Collaboration then typically occurs not only at the level of the source code, but also in mailing lists and via other means to discuss evolutions across modules, and is therefore more difficult to monitor. There are ways to map the dependencies in the code[17] and we definitely plan to take them into account in future research.

However, we found preliminary evidence of the role of cyclomatic (“McCabe”) code complexity in connexion to C-Mode. Although this definition of complexity does not take interdependencies into account, we still find correlations – at this descriptive statistical level, again – between C-Mode files and the existence of complex functions. This is particularly striking in GCC. An interesting question then is whether complexity might have increased *precisely* because C-Mode files have received many contributions from various maintainers, or whether it could be correlated to the existence of technical interdependencies for which it therefore be more or less a proxy of modularity. It remains however that complexity really seems to play a role that deserves further investigations.[19]

6. FURTHER WORK

This paper documented our first experiences with the exploration of detailed development records to study the allocation of tasks within virtual teams in open-source projects. The success that many of these projects have had in recent years and the voluntary nature of their development process make them extremely interesting to study, especially since abundant documentation of the development history of each project is readily available on the Internet. In this first attempt, we limited ourselves to the information that could be extracted from the logs that are kept by the version control systems that are widely used to keep track of changes to the software source code during the development process. So far, we have studied the logs of nine highly regarded open source projects: We have found patterns of task allocation in the records of code changes in these projects between files maintained by a single vs. multiple developers. We have suggested several potential explanations for these patterns. The extent to which these developers actually coordinate their code changes is not yet clear however. Further studies are needed to uncover the role played by various factors, and notably by complexity at various levels in the assignment of tasks between developers within open-source projects: Such studies would also allow to make further progress in the macro modelling of open-source communities at work, for which stigmergic models have recently been suggested.[20] Of particular interest here is the fact that some concentration of developer efforts on some specific files or modules seems necessary to explain the high concentration of module sizes observed empirically. The determinants of this concentration on “hot spots” have still to be understood, and this paper aimed also at making a further step in this direction. More generally, analysing the allocation of tasks within open-source projects would also allow us to learn more about actual development methodologies and notably about the governance of development activity, more in particular about the decision processes involved in the application of proposed changes to the code base.

ACKNOWLEDGMENTS

Our research has been supported by *Calibre*, a EU FP6 Coordination Action. JMD also gratefully acknowledges the support of the US National Science Foundation which has provided support to related work on open-source software. Finally, MdB would like to thank discussants at the ESSID summer school in Corsica plus John Gabriel Goddard and Héla Masmoudi in Paris for their suggestions and comments.

REFERENCES

1. C. U. Ciborra, *Teams, markets and systems*. Cambridge, UK: Cambridge University Press, 1993.
2. J. Olson and K. M. Branch, "Teams and project- and program-based organizations," in *Communication, Management Benchmark Study* (E. L. Malone, ed.), ch. 8, pp. 133–142, USA: Office of Science, Department of Energy, 2002.
3. W. van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters, "Workflow mining: A survey of issues and approaches," *Data & Knowledge Engineering*, **47**, pp. 237–267, 2003.
4. S. Koch and G. Schneider, "Effort, cooperation and coordination in an open source software project: Gnome," *Information Systems Journal*, **12**, no. 1, pp. 27–42, 2002.
5. J. Y. Moon and L. Sproull, "Essence of distributed work: The case of the Linux kernel," *First Monday*, **5**, 2000.
6. V. Valloppillil, "Open source software: A (new?) development methodology." Microsoft memo, 1998.
7. J. Matusow, S. McGibbon, and D. Rowe, "Shared source and open solutions: An e-government perspective," in *Proc. OSS* **1**, pp. 263–266, 2005
8. G. D. Prato and D. Gagliardi, "Proprietary software and open source philosophy: A shift in software's production methods," in *Proceedings of the 1st International Conference on Open Source Systems* (M. Scotto and G. Succi, eds.), (Genoa, Italy), pp. 237–240, 2005.
9. E. S. Raymond, "The cathedral and the bazaar," *First Monday*, **3**, March 1998.
10. S. Krishnamurthy, "Cave or community? An empirical investigation of 100 mature open source projects," *First Monday*, **6**, 2002.
11. K. Crowston and J. Howison, "The social structure of free and open source software development," *First Monday*, **10**, 2005.
12. W. Scacchi, "Understanding the requirements for developing open source software systems.," *IEE Proceedings – Software*, **149**, pp. 24–39, 2002.
13. A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," *ACM Transactions on Software Engineering and Methodology*, **11**, pp. 309–346, 2002.
14. M. Hahsler and S. Koch, "Discussion of a large-scale open source data collection methodology," in *Proc. HICSS* **38**, 2005.
15. G. Robles, S. Koch, and J. M. González Barahona, "Remote analysis and measurement of libre software systems by means of the CVSAnalY tool," in *Proc. ICSE* **2**, 2004.
16. B. Curtis, S. Sheppard, P. Milliman, M. Vorst, and T. Love, "Measuring the psychological complexity of software maintenance tasks with the Halstead and McCabe metrics," *IEEE Transactions on Software Engineering*, **5**, pp. 96–104, 1979.
17. A. MacCormack, J. Rusnak, and C. Baldwin, "Exploring the structure of complex software designs: An empirical study of open source and proprietary code," Working Paper 16, Harvard Business School, 2005.
18. J. Howison, M. Conklin, and K. Crowston, "OSSmole: A collaborative repository for FLOSS research data and analyses," in *Proceedings of the 1st International Conference on Open Source Systems* (M. Scotto and G. Succi, eds.), (Genoa, Italy), 2005.
19. A. Capiluppi, A. E. Faria, and J. F. Ramil, "Exploring the relationship between cumulative change and complexity in an open source system," in *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering (CSMR'05)*, 2005.
20. J.-M. Dalle and P. David, "Simulating code growth in libre (open-source) mode," in *The Economics of the Internet* (N. Curien and E. Brousseau, eds.), Cambridge University Press, 2005.
21. J.-M. Dalle and P. David, "The allocation of software development resources in 'open source' production mode," Discussion Paper 02-27, Stanford Institute for Economic Policy Research, Stanford, CA, 2003.

ANNEX

Sample extract from a CVS-log:

```
RCS file: /cvsroot/gaim/gaim/Attic/buddytrans.c,v
Working file: gaim/buddytrans.c
head: 1.3
branch:
locks: strict
access list:
keyword substitution: kv
total revisions: 3;     selected revisions: 3
description:
----------------------------
revision 1.3
date: 2001/04/27 21:51:09;  author: warmenhoven;  state: dead;  lines: +0 -0
these aren't necessary anymore because gaim can import winaim lists.
----------------------------
revision 1.2
date: 2000/04/05 08:22:38;  author: warmenhoven;  state: Exp;  lines: +1 -0
Made it very easy to switch between penguin and devil icons in the applet.
Also made it so that it would find the icons better (through use of nifty
GNOME functions I accidentally found). Other little touch-ups here and there.
----------------------------
revision 1.1
date: 2000/03/28 21:21:33;  author: warmenhoven;  state: Exp;
Translated buddytrans from perl to C. To be swallowed by gaim later.
```

Table 1: Summary description of the logs: first and last month of activity, total number of commits, of authors, of files involved, minimum and maximum number of commits, of authors, and of files per month over all months of activity.

	first	last	commit	min	max	authors	min	max	Files	min	max
gcc	08/97	08/05	241347	329	9540	250	5	105	34757	185	8730
NetBSD	03/93	08/05	224887	201	4688	267	3	78	19514	179	3657
gaim	03/00	08/05	37666	45	4207	39	2	24	5158	23	2123
mozilla	03/98	08/05	411235	922	16104	595	2	146	40545	661	14549
apache	07/96	11/04	57581	16	2881	79	4	31	4133	12	1491
python	08/90	09/05	62382	22	1645	88	2	32	4643	19	1107
cvs	12/94	08/05	24105	12	1581	30	2	11	1062	10	487
openssh	09/99	08/05	7483	2	403	50	2	12	289	2	242
postgresql	07/96	08/05	85023	73	2191	25	2	12	4102	57	1719

Table 2: Comparison between C and C++ files to which only one person applied changes (I-mode) and those files to which more than one person applied changes in the last 30 days of activity on the file (C-mode). The columns *I-mode* and *C-mode* list the total number of files in each set; the other columns indicate whether the average C-mode value is significantly different than the average I-mode value for one metric per column for various metrics: *sloc* – the number of lines of code in a file, *McCb* – the maximum McCabe complexity of functions in a file, *funs* – the number of functions in a file, *mons* – the number of months between the first and the last revision on the file, *revs* – the number of revisions on a file, *auth* – the number of committers to a file, and *addM* – the number of lines of code added to a file during the last 30 days of activity. The number of stars indicates the level of confidence of the two-sided t-test – 95% for one, 99% for two, and 99.9% for three stars.

	I-mode	C-mode	sloc	McCb	funs	mons	revs	auth	addM
gcc	510	657	***	***	***	***	***	***	**
NetBSD	291	341	***	***	***	***	***	***	
gaim	68	79			**	***	***	***	***
mozilla	501	506		***		***	***	***	
apache	129	80	**	**	*		**	***	
python	36	26	***				*	*	
ccvs	54	11				*		*	
openssh	16	16							
postgresql	141	95							

Table 3: Confidence level of one-sided t-tests on the same sets of files as in Table 2 for the same metrics. Black stars (justified on the right) indicate a significantly higher value for the C-mode and red stars (justified on the left) indicate a significantly higher value for the files in the I-mode set according to the column metric. The number of stars still indicates the level of confidence of each of the 2 one-sided t-tests – 95% for one, 99% for two, and 99.9% for three stars.

	I-mode	C-mode	sloc	McCb	funs	mons	revs	auth	addM
gcc	510	657	***	***	***	***	***	***	**
NetBSD	291	341	***	***	***	***	***	***	
gaim	68	79	*	*	***	***	***	***	***
mozilla	501	506		***	*	***	***	***	*
apache	129	80	**	**	**		**	***	
python	36	26	***			*	*	**	
cvs	54	11	*			**		*	
openssh	16	16							
postgresql	141	95							

Table 4: I-mode averages (same metrics).

	n	sloc	McCb	funs	mons	revs	auth	addM
gcc	510	449.5412	53.63922	20.51764	6.37662	3.668627	1.509804	21.21176
NetBSD	291	7718.855	54.76632	34.13058	45.87331	15.28866	5.402062	26.42612
gaim	68	12664.35	16.01471	24.95588	15.86422	20.08823	3.455882	18.95588
mozilla	501	3384.279	172.1656	71.45109	35.54785	24.20359	8.085828	51.10180
apache	129	642.7132	56.58915	11.72093	38.21014	34.53488	7.968992	59.13178
python	36	26588.38	179.7777	55.08333	60.60129	38.83333	6.166667	44.33333
cvs	54	1311.074	23.51852	9.074074	63.81995	45.77777	5.166667	49.09259
openssh	16	13476.62	153.6875	52.18750	51.46588	77.56250	7.812500	20.56250
postgresql	141	1391.390	140.6666	22.31914	72.46417	88.87943	6.248227	100.1063

Table 5: C-mode averages (same metrics).

	n	sloc	McCb	funs	mons	revs	auth	addM
Gcc	657	4334.958	348.3485	83.55251	31.27481	68.02892	13.08067	43.80061
NetBSD	341	9694.915	71.00000	43.06158	59.72404	28.34311	9.395894	28.59531
gaim	79	4557.050	28.16456	42.51898	26.22441	103.4683	8.658228	86.88608
mozilla	506	3303.247	273.8379	101.7964	50.19817	80.93281	21.24308	68.61858
apache	80	954.4375	104.1000	17.35000	36.27420	60.95000	11.38750	65.56250
python	26	1635.538	254.3076	41.53846	91.11097	96.19231	11.88461	43.53846
cvs	11	1735.545	22.45455	9.727273	104.9525	52.90909	8.000000	43.45455
openssh	16	13259.87	205.4375	60.68750	60.65967	82.18750	9.062500	25.00000
postgresql	95	1607.463	153.7473	26.22105	63.40973	80.33684	5.747368	96.52632

Systematic measurement of perceptual design qualities

Michael Bittermann and Özer Ciftcioglu‡

[a] Chair of Technical Design & Informatics, Architecture Faculty, Delft Technical University, The Netherlands
[b] Chair of Technical Design & Informatics, Architecture Faculty, Delft Technical University, The Netherlands
M.Bittermann@bk.tudelft.nl

ABSTRACT

Implications of design decisions are hard to oversee for designers. This is the case in particular with respect to decisions, which influence perception related qualities of designs. Such qualities are for example visual openness, visual privacy, and spatial intimacy. They are difficult to measure because of their subjective and soft nature. Measurements of such qualities are important because they are basis for user-oriented, optimal decisions in architectural design. Existing attempts in the architectural domain to assess such qualities systematically are not based on suitable models of space perception. Their ability to assess perception aspects of designs is limited. In this paper a new real-time measurement system for design is presented, which is based on a computational model of visual space perception. Core method of the perception model is a new method of geometric analyses termed Random Direction Distance Sampling (RDDS). Core method of the measurement system is exponential averaging, which is a time-series analyses method from the domain of Signal Processing known as exponential averaging.

Keywords: spatial perception, computational perception modelling, computational design, design assessment, design measurements

---

‡ Senior author

1. INTRODUCTION

Design is a complex process. This complexity stems from ill-defined, and time varying design requirements, as well as voluminous solution space. Design requirements generally include requirements for perceptual design qualities. Systematic assessment of such qualities is a traditional bottleneck in design, in particular in architecture and interior design. Such qualities are visual openness, visual privacy, spatial intimacy and geometric variance. Assessment of such qualities is imperative to evaluate the satisfaction of design requirements, which is an essential component in design optimization. Satisfaction assessment outcomes guide the search for optimal design solutions. Real-time provision of measurements is rather imperative to ensure efficient and effective optimality search, and to allow real-time adjustment of requirements in course of design. The central question addressed in this paper is the following.

How can perceptual qualities of designs be measured in real-time?

Existing attempts in the architectural domain to assess perceptual qualities are not based on suitable models of visual space perception. They are generally based on conventional computations of spatial component information such as relative amount of openings in spatial enclosures (Koile, 1997, Franz et. al, 2005). The methods in use are generally based on Isovist analyses or analyses of graph theoretic design representations (Hillier et al., 1984). Isovists, which were introduced by Benedict in 1979 (Benedict and Burnham, 1981), are polygons, which enclose the volume directly visible from a location within a space. In many applications they were further simplified to the horizontal slices of these volumes at eye-height. Using Isovists for spatial analyses, certain properties of visual space perception are strongly simplified or not considered at all. These properties are variance in the significance of spatial directions in the visual field, detailed analyses of the geometric variance in the spatial envelope, and transition conditions, such as motion of the perceiver or real-time modifications of the spatial environment during design. Graph theoretic design representations are representations of designs in which design elements are represented as nodes, which are linked with other nodes in a network structure. Graph properties, such as mean shortest path length, etc. can be identified. Such graph analyses results are considered to be correlated with certain perceptual qualities (Turner et al., 2001), however, graphs identify visible locations only indirectly, via a certain network grid, and not directly in terms of physical visibility. Both, Isovist and graph based approaches are not based on modelling the visual space perception process. Due to sensitivity of visual space perception regarding the constitution of the visual field, detailed geometric properties of space, as well as transitions conditions, their ability to assess perceptual design qualities is limited. Real-time provision of measurement outcomes is rather imperative due to sensitivity of optimality searches with respect to simultaneous availability of all relevant design assessment information. In this paper a new real-time measurement system for design is presented, which is based on a computational model of visual space perception. Before coming to the explanation of the perception model and the measurement system, firstly the perceptual qualities measured in this research are concisely defined as follows.

2. DEFINITIONS

2.1 Perceptual Qualities

Perceptual qualities, such as visual openness, visual privacy, and spatial intimacy are qualities inherent to a design, which influence the perception of the design product. They are inherent to the design, because their existence and constitution is immanently linked to the existence and constitution of the design. The inherence is valid provided that the geometric constitution of the design remains unchanged. Because they are inherent to a design, they can be assessed during the design in place of afterwards, and they can be verified afterwards if that would be so desired.

2.2 Visual Openness

Visual openness is an inherent quality of a design, which describes how much a design allows for visual perception of distant positions. In the following, firstly the definition of visual openness of a single geometric position in a design is given, secondly the visual openness of a design as a whole is defined. The visual openness of a position in a design describes, how much a design permits retrieval of visual data from distant positions. Generally, opaque elements, such as walls, columns, furniture, etc. prevent visual perception of distant positions. The visual openness of a design is combined visual openness information coming from visual openness assessments of a number of relevant positions in a design.

2.3 Visual Privacy

Privacy, in general, is the ability of an individual to govern availability of hid/her information. As an inherent quality of an architectural design, privacy indicates, how much a design enables government of availability of information. Information of concern, in context of architectural design, is primarily of acoustic and visual nature. Privacy in architecture is consequently including visual privacy and acoustic privacy. Visual privacy of a position in a design describes, how much a design prevents retrieval of visual data of that position from other positions in the design. Generally, opaque elements, such as walls, columns, furniture, etc. prevent visual data retrieval from positions in the environment surrounding the position. The visual privacy of a design is combined visual privacy information coming from visual privacy assessments of a number of relevant positions in a design.

2.4 Spatial Intimacy

Spatial intimacy is an inherent quality of a design. Spatial intimacy of a position in a design describes, how much that position is enclosed by nearby opaque objects. The spatial intimacy of a design is the combined spatial intimacy information coming from spatial intimacy assessments of a number of relevant positions in a design.

2.5 Geometric Variance

Perceptual geometric variance is defined here as the exponential average of the variance of perception samples, which are variance of geometric distance samples. The greater the variance of the perception samples, the greater the perceived geometric variance. As consequence of this definition, the geometry perceived as most simple is the interior of a sphere perceived from its centre position. An example of a space with a high perceived geometric variance is the centre of a treetop, where distances to individual opaque points strongly vary.

3. MODELLING VISUAL SPACE PERCEPTION

In line with Helmholtz' definition of vision as a form of unconscious inference, which is defined as a process of deriving probable interpretation for incomplete data (Wade, 2000), and Marr's definition of vision as a process of information processing and representation (Marr, 1982), visual space perception can be defined as the retrieval and processing of visual data from the environment surrounding a perceiver. It is assumed that distances of positions surrounding a perceiver can be obtained accurately by visual perception. Sufficient accuracy of visual distance retrieval under normal spatial circumstances, with a large number visual depth cues in the perceived environment is assumed. The perception model presented here is a cyclopean model, which means the visual apparatus is represented with a single geometric point. Essential perception task is continuous retrieval of distance-information coming from positions surrounding the perception position. This process is termed Random Direction Distance Sampling (RDDS) here. Initial motor of the model is continuous generation of sightlines in random directions. Three uniform random numbers are used as components of a 3-dimensional direction vector. The utilization of random vectors as source for the directions of the sightlines is imperative due to unpredictability of geometric constitution of the measured environment, in particular with respect to the scale of its geometric roughness. The vector components of the direction of each sightline are shaped by means of Gaussian shape filtering in the following form.

$$f(x) = \left(x_{source} + x_{mean}\right) \times \sqrt{x_{dev}} \tag{1}$$

By means of modifying the parameters of the Gaussian normal distribution, x_{mean} and x_{dev}, the visual field, that is the probabilistic distribution of the orientation in sightline generation, can be adjusted in real-time.

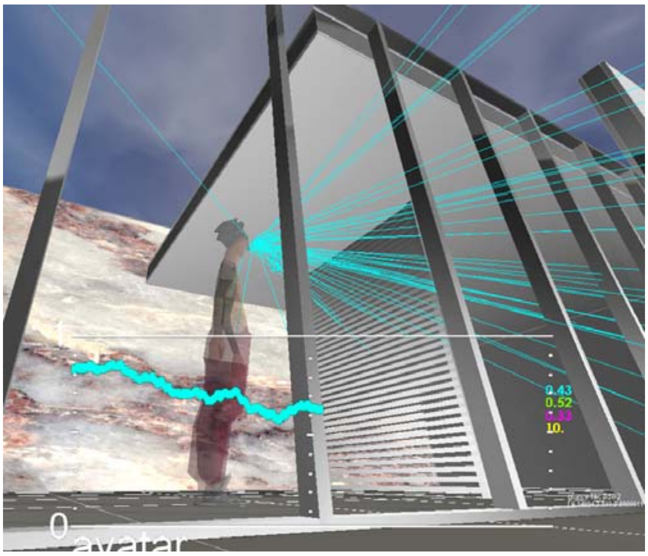


Figure1. Display of modelled visual space perception. Here the sightlines used to obtain the current measurement are shown, as well as the graph plot, which gives the measurement outcome.

A number of sightlines visually form a cone, with a greater density of sightlines in the centre of the cone and a reduced density of sightlines in the periphery of the cone, in line with the phenomenon of focal sharpness/blurriness in visual space perception (see Figure 1). Any geometry of visual cone can be achieved by means of the parametric adjustment of the shape filter just mentioned. Each sightline delivers an individual data-sample when intersecting surrounding geometry. These samples are continuously processed by means of weight filters. The weight filters are defined in accordance with the definition of the spatial quality to be measured. In particular the relation of distances *x* obtained by perception rays with respect to the quality to be measured expressed in the functions of the weight filters. In visual openness measurement, preliminary a sigmoid-based function is used as weight filter.

$$S = \frac{1}{1 + e^{-\left(x - \frac{l_{\max} - l_{\min}}{2}\right)}} \tag{2}$$

In visual privacy and spatial intimacy measurement, preliminarily Butterworth function is used.

$$S = \frac{1}{1 + \left(\frac{x}{l}\right)^{m}} \tag{3}$$

The function parameters can be modified to adjust the measurement calibration to match with different definitions measurement conditions. Alternatively to Sigmoid and Butterworth function, weight-filters based on fuzzy membership functions can be used. Thereby the weight-filtering can be controlled in a more detailed way, to match any non-linearity of distance-based perceptual quality definition. In particular the phenomenon of distortion of the Mueller-Vieth horopter in visual perception, known as Hering-Hillebrand deviation (Howard and Rogers, 2002), can be taken into account in a fuzzy model. Details of the fuzzy modelling methodology are beyond the scope of this paper.

4. FROM PERCEPTION TO MEASUREMENT

4.1 Time series analyses

The weight-filtered samples are analyzed by means of a time-series analyses method borrowed from the domain of Signal Processing. This method is known as exponential averaging. Exponential averaging identifies average signal-values by means of continuous weight filtering of signal values using a time constant τ in this form.

$$\omega \equiv 1 - \frac{1}{\tau} \tag{4}$$

The time constant represents the size of a time-window in which samples are averaged. The time window moves forward in time, which corresponds to continuous update of the average value, which is incorporation of one new sample and dropping the latest sample at each time-step. At each time step q the new exponential average P_q of the signal S_q is computed in this form.

$$P_q = \omega P_{q-1} + (1-\omega)S_q \tag{5}$$

Contrasting conventional averaging methods, in exponential averaging previously obtained average information is incorporated in computation of the current average. This way the average, which is the measurement outcome, is updated in real-time in a computationally efficient and effective way (Ciftcioglu and Peeters, 1995).

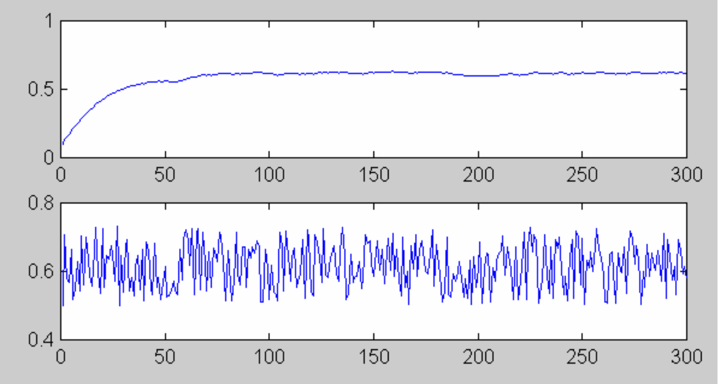


Figure 2. Example of a measurement. Below weight filtered perception samples S. Above exponential average P.

After some time, with no changes to the measurement system, the measurement stabilizes at a certain value (see Figure 2). Proportional to the time constant this stabilization is more or less quickly establishing, and is more or less stabile. In situations of static measurement conditions, a greater time constant yields a more accurate measurement. Latency effects of exponential averaging in transition conditions apparently correspond to the phenomenon of spatial memory in visual space perception. Transition conditions are changes in the measurement system over time, in our case in the form of translation and/or rotation of the cyclopean eye of the measurement system or modifications of the geometry surrounding the eye. Establishment of new average values in such transition conditions take time, which is termed measurement latency in this context. This latency is proportional to the time constant provided constant processing frame rate. The following example serves to illustrate the significance of this inherent property of exponential averaging in relation to visual space perception, in particular to the phenomenon of spatial memory in transition conditions. A person looking around or moving through an environment generally receives different openness impressions at each instance. However the previous impressions are not forgotten immediately but remain in the consciousness of the perceiver (Baddley, Logie..). This effect appears to correspond to the latency effect of exponential averaging method presented earlier. Greater values for τ correspond to greater spatial memory.

4.2 The Concept of Memory Time in Perception

In attempts of modelling real-time visual space perception, which is matching of time durations for establishment of computational perception assessments and human perception, another property of real-time systems has to be considered, which is expressed in the concept of constant memory-time. According to this concept, which is borrowed from the domain of real-time systems engineering, multiplication of time-constant τ and the reciprocal value of the computational frame-rate are a constant number termed memory-time μ.

$$\mu = \tau \times \frac{1}{f} \tag{6}$$

Thereby, variation of computational frame-rate *f* is reflected in the measurement in the form of variation of the time constant. Slower computational processes, which are processes with a lower computational frame-rate, imply smaller time-constants and vice-versa. Experimental verification of the system variables can be conducted, which is beyond the scope of this paper. The parameters of the Gaussian shape filter, the weight-filter functions and the time-constant of the exponential averaging can be modified in real-time. Thereby the measurement system can be calibrated based on differently shaped visual perception, particular definitions of visual openness, visual privacy, and spatial intimacy, and individual difference in spatial memory and computational frame-rate.

4.3 Measuring Geometric Variance

Perceptual geometric complexity measurement, contrasting the measurements of the other perceptual qualities, is defined in this context as the exponential average of the variance of the distance samples. The complexity measurement is done by variance computation of the perception signal P in this form.

$$C_q = \omega \times C_{q-1} + (1 - w) \times x^2 - P_q{}^2 \tag{6}$$

The greater the variance of the perception samples, the greater the perceived geometric variance. As consequence of this definition, the geometry perceived as most simple is the interior of a sphere perceived from its centre position. An example of a space with a high perceived geometric variance is the centre of a tree top, where distances to individual opaque points strongly vary.

5. EXPERIMENT

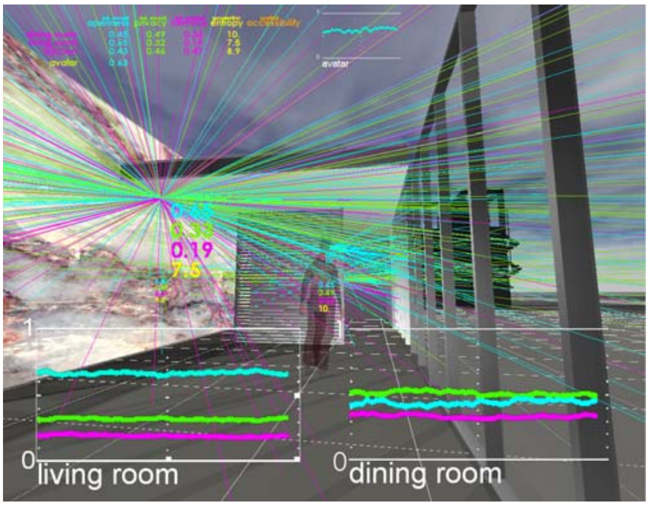


Figure 3. Real-time measurement of perceptual qualities in a design

In the following, a number of measurements are presented which were taken from several positions in a design. The design contained a number of geometric details, such as facade studs, screens made from horizontal louvers, windows doors, etc. Above Position 1 no ceiling is designed yet, above all other positions there is a ceiling located at 2,65m height.

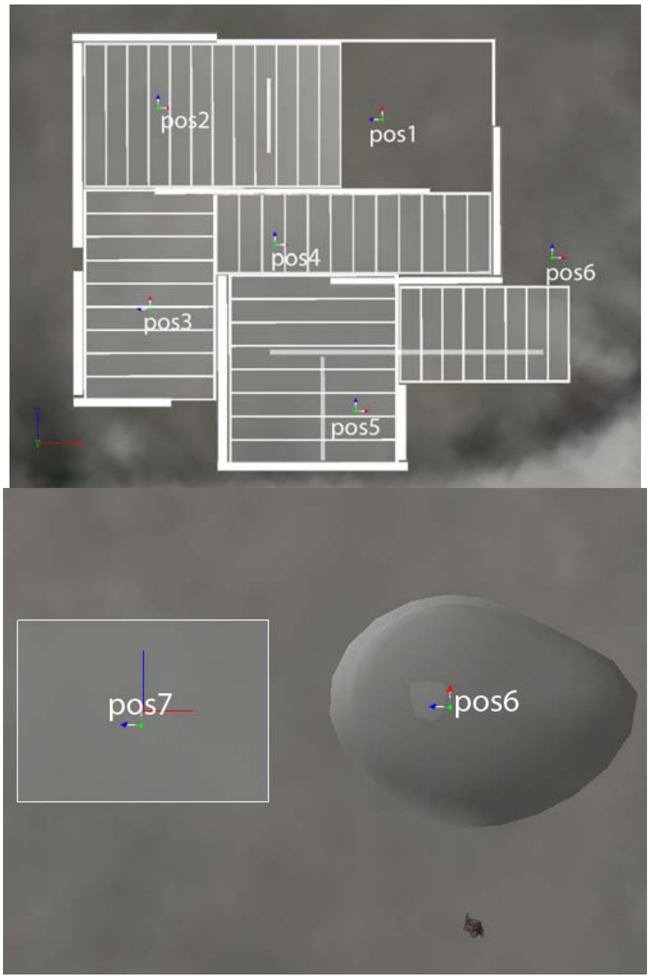


Figure 4. Plan view of measurement positions

The locations of the measurement positions were as shown in figure 4. Height of the measurement positions was chosen to be 1,70 above floor height, which is around average eye height. The parameter settings for the measurement system were as follows.

Table1. Parameter settings of measurement system for experiments

	visual openness	**visual privacy**	**spatial intimacy**	**geometric variance**
Shape filter				
$(x_{mean,} x_{dev})$	(0.00, 1.00)	(0.00, 1.00)	(0.00, 1.00)	(0.00, 1.00)
$(y_{mean,} y_{dev})$	(0.00, 0.02)	(0.00, 0.01)	(0.00, 0.50)	(0.00, 0.01)
$(z_{mean,} z_{dev})$	(0.00, 1.00)	(0.00, 1.00)	(0.00, 1.00)	(0.00, 1.00)
Weight filter				
function	sigmoid	Butterworth	Butterworth	Butterworth
lmax	4.5	7.5	4.0	7.5
lmin	1.8	3.0	1.8	3.0
m	-	2.0	3.0	2.0
Exp. averaging				
time constant τ	400	400	400	200

The shape filter was set to simulate 360 degree perception in the horizontal plane with some divergence in vertical direction off that plane (see y_{mean} , y_{dev} in Table 1). The setting of the weight filter parameters is suitable for interior measurements, based on previous tentative experiments. The time constant setting for the experiments is τ =400 and τ =200. The exact setting of the time constant is irrelevant in this experiment since transition conditions were not involved, real-time perception was not considered and the measurement was obtained after stabilization of the measurement outcome. With a high value for the time constant, after sufficient measurement time, the measurement stabilizes at a certain number. This number is the measured value of the perceptual quality.

6. RESULTS

The measurement outcomes for the measurements based on the system settings given in Table 1 and the positions indicated in figure 2 are as follows.

Table 2. Measurement outcomes

	visual openness	**visual privacy**	**spatial intimacy**	**geometric variance**
Position 1	0.72	0.34	0.19	6.6
Position 2	0.50	0.48	0.33	7.6
Position 3	0.48	0.50	0.29	10
Position 4	0,52	0,49	0,36	12
Position 5	0,17	0,80	0,61	1,6
Position 6	0,73	0,33	0,21	5,3
Position 7	0,30	0,65	0,48	4,6
Position 8	0,30	0,65	0,52	4,8

The outcomes presented in Table 2 are normalized since the weight filtering delivers normalized output (see formulae 2 and 3). Based on preliminary experiments the measured values appear to be in accordance with expectations based on visual inspection of the design, both in plan and from 1st person perspective. Small deviation in personal judgment may be due to imprecision in judgment. Larger and in particular structural difference in personal judgment and measured values may originate from various source. An individual may have a particular shape of visual perception or a particular definition of the weight-filtering parameters. Experimental identification of such difference can be conducted and thereafter knowledge-based calibration of the measurement system is possible by means of evolutionary search algorithms, which are a methodology from the domain of Computational Intelligence. This methodology can systematically find those settings of measurement system parameters, which yield minimal deviation to experimental data obtained by statements of test persons. Details of this procedure and methodology are beyond the scope of this paper. Measurement latency of exponential averaging, mentioned earlier, can be reduced by incorporation of Kalman filtering for smoothing in the measurement system, which is a sophisticated modelling methodology from the domain of Signal Processing. Via Kalman filtering the establishment of average signal values is obtained faster than via exponential averaging alone. Kalman filtering essentially models and eliminates process noise. Details of the method are beyond the scope of this paper. A particularly interesting result is the geometric variance assessment of position 7 and 8. Here the interior of a blob and a box geometry are measured. Although blobs are often considered to have a complex geometry with string variance in curvature, this perception based measurement reveals that in fact the blob geometry has a rather low geometric variance, very similar to the measured variance of a box. From a perception viewpoint both geometries differ only slightly in their geometric variance. This is not surprising considering that blobs are generally geometries with strong affinity to the sphere, which is the geometry with least variance according to our definition. Detailed analyses of the variance measurement by means of wavelet analyses, which is an advanced signal analyses methodology from the domain of Signal Processing, indicate that there is notable difference in the characteristics of the individual composition of blob and box geometry concerning their variance signals. This is due to difference in smoothness/edginess in both geometries.

The outcomes of the measurements are plotted in graph form in real-time. This way the designer has real-time feedback concerning the implications of his design actions with respect to the perceptual qualities, which are measured. The outcomes are used to assess requirement satisfaction in real-time. A number of demands are expressed in the form of required values for individual qualities and respective tolerances. The tolerances are translated to weight-factors by means of simple fuzzy membership functions. Individual deviation from demands is thereby weighted according to relative importance among requirements. Satisfaction of the overall design requirement is assessed in real-time. This outcome can serve as fitness assessment in computational design optimization.

7. CONCLUSIONS

The real-time measurement system presented in this paper is able to measure perceptual qualities in real-time, based on the definitions given. The measurement system can be calibrated in real-time to match individually different visual perception as well as individual definitions of perceptual qualities. The system deals with any detail level of geometric detail in the design. It provides measurements in real-time and handles transition conditions. Therefore it is particularly suitable in design, where fast response to design modifications is rather imperative for optimality search. Methodologies from the domain of Signal Processing, in particular exponential averaging, are suitable for computational perception modelling in particular perception based design assessment. The setting of the time-constant in exponential averaging can be adjusted in real-time to model time-based visual space perception, which introduces the concept of memory-time from systems engineering to architecture. The measurement outcomes form essential contribution in holistic requirement satisfaction assessment by resolving a traditional bottleneck in computational design, which is to deal with perception related design qualities. Such assessment is imperative basis for systematic optimality search, which is the essential process in design.

REFERENCES

1. Benedikt M L, Burnham C A (1985), ``Perceiving architectural space: from optic rays to isovists", *Persistence and Change* Eds W H Warren, R E Shaw, Lawrence Erlbaum Associates, London
2. Ciftcioglu O, Peeters T.T. J. M. (1995) "Statistics on Exponential Averaging of Periodigrams." IEEE Trans. on Signal Processing, Vol.43, No.7
3. Franz G M, von der Heyde and Bülthoff H (2005) “Predicting experiential qualities of architecture by its spatial properties” *Designing Social Innovation: Planning, Building, Evaluating*, 157-166. (Eds.) Martens, B., A. Keul, Hogrefe and Huber, Cambridge, MA (2005)
4. Hillier, B.; Hanson, J. (1984) *The Social Logic of Space*, Cambridge University Press
5. Howard I, Rogers B (2002) “Depth perception“, *Seeing in Depth*, Vol. 2, Porteous
6. Koile K. (1997) “Design conversations with your computer: Evaluating experiential Qualities of Physical Form” Junge, R. (ed.), Conf. Proc. 7th Int. Conf. on CAAD, in Munich, Germany. Vol.
7. Marr D. (1982) *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H. Freeman and Company, NY 29-61
8. Wade N,J (2000*) 'Introduction', Helmholtz's Treatise on Physiological Optics*, Bristol: Thoemmes Press, , pp. v - xxi
9. Turner A, Doxa M, O’Sullivan O, Penn A (2001) “From isovists to visibility graphs: a methodology for the analysis of architectural space” Environment and Planning B: Planning and Design volume 28, pages 103 ^ 121

The representation of common ground and its role in P2P-supported design team processes

Michael Cumming [a] and Evren Akar [b]

[a] Faculty of Architecture, Delft University of Technology, Berlageweg 1, Delft, Netherlands
[b] Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, Delft, Netherlands
m.cumming@bk.tudelft.nl

ABSTRACT

The concept of common ground (CG), found within social linguistics and other domains concerns the contributions to mutual knowledge, mutual beliefs, and mutual assumptions that inform social and collaborative activity. Construction of mutual knowledge through contributions to CG is necessary for people to make logical inferences about what their interlocutors know. Without access to such information, exchanges found within conversations would be impossible to disambiguate. CG representation and management is seen as necessary for the management of distributed, often quite ambiguous, collaborative design processes. Online peergroups found within peer-to-peer (P2P) systems can support and represent types of communities. Such peergroups have the potential, unlike non-computer-mediated social groups, of being able to explicitly represent CG as it dynamically emerges in practice. This paper explores the implication of dynamically representing emergent CG and discusses how such representations can support collaborative design processes. Details are provided about how this can be implemented within P2P-based design coordination applications.

Keywords: Common ground, grounding, emergence, peer-to-peer (P2P).

BACKGROUND

Definition of common ground

CG is required for the comprehension of normal conversational interactions. It is also essential for the coordination of joint actions of all sorts. Clark's central thesis that "language use is really a form of joint action", i.e., action carried out by an ensemble of people acting in coordination with one another (Walker, 1997). Joint activities require coordination of both the content of the activity and the process by which the activity moves forward. According to Clark, "…to coordinate, we have to appeal, ultimately, to our current common ground. At the same time, with every joint action he and I perform, we add to our common ground. This is how joint activity, from chess games to business transactions, progress. When my son and I enter a conversation, we presuppose certain common ground, and with each joint action–each utterance, for example–we try to add to it. To do that, we need to keep track of our common ground as it accumulates increment by increment" (Clark, 1996: 92).

Relevance of CG to collaborative design

Within the design research context, CG tends to be used more metaphorically and informally than in linguistics and usually refers to the common understandings that designers bring to a design process and how these understandings accumulate within collaborative design teams as designers learn to work together.

Some researchers, however, consider design collaboration from a 'language as action' perspective (Flores, Graves, Hartfield, & Winograd, 1992) and see it as a specialized form of conversation, in which collaborators define and refine group identity and strategy "using conversational turns to display their understanding of the current state of activity, an understanding that other participants may, in subsequent conversational turns, either ratify or correct. Through sequences of such conversational pairs, participants accumulate the common ground necessary to support common goals" (Geisler & Rogers, 2000: 398). According to Geisler and Rogers, collaborative design has specialized rules for the accumulation of CG. Due to the multiple perspectives that multi-disciplinary design work involves, there is the implication that people cannot share completely their understanding of a design process with people from other disciplines. There must be both a shared and a private discourse in which the shared portion is used to solve problems of group concern, while that within a specific discipline handles problems that may not be of practical concern to other disciplines. "In multidisciplinary design contexts, then, the normal rules for the accumulation of CG must be relaxed or otherwise modified so that designers become willing to ratify proposals of which they do not have complete understanding." (Geisler & Rogers, 2000: 398). Fischer notes that design communities such as those found on collaborative design teams, must bridge differences arising from temporal, technological, and conceptual distances. Bridging conceptual differences requires support for the accumulation of CG and of shared understandings in general (Fischer, 2004). The stronger the CG is in a design team, presumably, the greater is the chance that team members will understand the meaning and context of design ideas, and be able to interact successfully on a social and technical level. Fischer also notes that design teams are types of Communities of Interest (CoIs) bring together stakeholders from different CoPs to solve a particular (design) problem of common concern (Fischer, 2004: 156). "In CoIs, boundary objects support communication across the boundaries of different knowledge systems, helping people from different backgrounds and perspectives to communicate and to build common ground" (Fischer, 2004: 156).

Participatory collaboration

According to Clark, construction of meaning in normal interactive conversations is a participatory collaboration, requiring both presentation and acceptance phases (Clark & Brennan, 1991). For example, one actor, A, presents an utterance, with the expectation that during the acceptance phase B will provide evidence that B understood the utterance. Contributions to CG are not assumed to take place simply because social interactions occur, but require explicit indicators and real-time acts of recognition from interlocutors. These indicators provide explicit evidence that messages are getting through, that meanings are understood, and that some kind of accumulation of shared knowledge is occurring. The CG-based model of conversational interaction depends on an interactive, incremental, and distributed view of knowledge production. It is within these distributed processes that complex design processes and products emerge.

Communal and personally experienced knowledge

There are two types of knowledge that inform conversational interactions: 1. That which is grounded by ones direct experience in interpersonal interactions, 2. That which is derived through knowledge of a culture, social community, or community of practice (CoP) (Wenger, 1999). Having in-depth knowledge of a culture's norms and structures enables one to make suppositions about what an agent is likely to know, and about which references he is likely to understand. These suppositions though, if they are to rise above the level of received opinion and cultural stereotype, ultimately must be confirmed by positive evidence within specific conversational exchanges.

All types of evidence, both grounded and ungrounded, can affect social behaviours and structures. Yet, as Clark describes, some forms of evidence can be perceived to have a higher level of quality than others (Clark, 1996: 98). This notion of quality assumes that evidence derived from verbal interactions has a degree of ambiguity and that high quality evidence has lower levels of ambiguity and a higher likelihood of being mutually perceptible within a conversational group. Given this, it is proposed here that CG grounded by personal experience tends to be of a higher quality than that which is received via cultural stereotypes. This may be especially true within collaborative design processes in which the cultural norms of various diverse design disciplines may work to align expectations, but may contribute little to informing the actual content of specific processes.

CG as a distributed entity between collaborators

In a computational or representational environment, in order to create notions of social structure, there must be some process of defining these social norms or expectations. As the history of cognitive and social modelling demonstrates, the top-down, analytically based definition of group norms tends to be difficult or impossible, and may also be empirically unjustified (Orlikowski & Yates, 1998). Collaborative design teams, when the participants are socially diverse, are a type of synthetic society. In new types of social groups there may be few relevant historical precedents, and it may be unclear whether existing social stereotypes are relevant, useful, or informative. In such situations, in which a variety of social agents are thrown together into novel configurations, it is usually not clear what complex, self-organizing effects of design processes and products might emerge (Schelling, 1978).

Structuration theory (ST) acknowledges the important role of 1. Institutions, …with their hierarchies and rules etc., 2. Group structures, with their role differentiation, values and norms, and 3. Individual characteristics, such as knowledge, skills, attitudes, and dispositions (Andriessen, 2002: 48). CG construction, at the level of the speech act, is at the bottom rung of a wide variety of interactions individuals may have, not only with other individuals, but also with other social groups and institutions. Therefore, accumulation of CG through direct social interactions is seen as one of the most prominent, and most fundamental means of creating social structure of various types.

The basic idea of CG is that it is a type of shared, or communal knowledge, which can be confirmed by complex signalling behaviours between interlocutors. However, this doesn't mean that CG resides in a common place accessible to all parties, in the metaphor of a shared database. Instead, each party of a communication act must have access to this shared knowledge through their own private cognitive resources – i.e., within their own head. Therefore, CG gives the impression of being shared and common, even though its content is represented in a distributed fashion. Because of such considerations, the approach here is to model the construction of social structure from the bottom-up, based on the linguistic notion of emergent CG that must be incrementally grounded within simple social interactions.

COMMON GROUND AND DESIGN PROCESSES

Common ground as key to design coordination

Clark proposes that CG is essential to coordination of joint actions and suggests that the shared basis for CG plays a crucial role in that coordination (1996: 94). For example, if an engineer and an architect agree to modify a specific column in a building, there must be belief, which is held by both parties that there is an agreement to modify a specific column, and that the nature of the modification to be made is also understood by both parties. Without this knowledge, about what the action involves and what the action pertains to a particular column, then there can be no coordination of the actions of the engineer and the architect. This knowledge must be shared: that is, both parties must believe a consensus exists, and this CG must be confirmed by the behaviours of both. Therefore, management of CG is an essential aspect of collaborative design processes. In order for collaborative work to proceed, and for designers to adequately coordinate their activities, designers must keep track of their shared knowledge as it incrementally develops within the design team. Designers must construct their CG both intentionally, as they attempt to address and solve specific design problems, and more spontaneously and unplanned, as they learn to work together as a social group.

Communal and personal knowledge sources

In collaborative design, knowledge, both culturally received and personally experienced, plays a role. Culturally received, or communal knowledge is that which design participants can assume others share, given the general cultural situation of the design process. For instance, for architectural design processes, participants might assume that others will likely understand the norms of standard design representations such as plans and elevations, and have an appreciation of basic structural and constructional principles (Blau, 1984) (Gutman, 1988). Received knowledge, such as ‘all architects know how to read building plans’ tends to be a useful generalization, but also can be disconfirmed, since it is not difficult to conceive of an architect who doesn’t know how to read plans. Therefore, knowledge that is derived from a collective stereotype and then applied to a specific instance, can be presumed, but still requires confirmation within personal interactions.

Processes of grounding

Grounding processes found in the literature

There are many references found in the literature regarding theories of grounding and social interactions based on H. H. Clark’s approach. These tend to congregate in domains concerned with: General linguistics: (Clark & Brennan, 1991) (Clark, 1996); Computational linguistics and conversational analysis: (Traum, 1994; Traum & Allen, 1994); General design theory: (Bucciarelli, 1994, 2003) (Schön, 1983); Collaborative design teams and their behaviour: (Larsson, 2003) (Fischer, 2004) (Geisler & Rogers, 2000) (Hendry, 2004); Design of CSCW systems: (Giboin, 1998) (Hoadley & Kilner, 2005) (Ure, Lloyd, Pooley, & Dewar, 2003); Ethnography and information ecologies: (Nardi, Whittaker, & Bradner, 2000) (B. Nardi & O'Day, 1999) (Star & Ruhleder, 1994) (Bowker & Star, 1999).

Incremental contributions in CG

CG, as presented by Clark, is an inherently bottom-up process: in order to make sense of conversations and to know what counts as relevant contributions to a conversation, it is essential to accumulate CG. CG is not an entity that is received from some source fully formed and ready to use. Instead, it must be built up layer by layer by active participants. CG requires a continuous process of grounding, i.e. using mechanisms to ensure B understands A’s utterances and to inform A about this understanding. Many mechanisms are used in conversation for this purpose, such as:

- *Acknowledgement*: ‘uh huh’, ‘yeh’, ‘mm’ etc.
- *Relevant next turn*: B gives a reaction that shows his/her understanding
- *Attention*: looking at the sender
- *Indicative gestures*: pointing by B at the object A is referring to
- *Verbatim repetition*, or spelling in the case of transferring numbers or names

These mechanisms are governed by what Clark and Brennan term the *principle of least collaborative effort*: participants in conversation try to minimise their collaborative effort for communication and that they seek to mutually reduce the amount of effort each must put into formulating new contributions, and interpreting others' contributions (Birnholz, Finholt, Horn, & Bae, 2005: 23). Three types of information can be used to support such grounding processes: 1. *Non-verbal information*: the non-verbal signals, such as gaze, intonation, and gestures, 2. *Object information*: the objects of the interaction, such as documents, images, or design representations, 3. *Context awareness*: the situation, i.e. the physical background and the setting in which the participants are communicating, may convey information about who does or does not participate in the interaction, about the work context, or about the status of the participants.

CG accumulates as people interact. The relationships that they build with and build is based on what they perceive are shared between them. Each step builds upon the step before. When people hold a conversation, they may bring a perceived CG with them before they interact. And with each joint action, such as design discussion or meeting, they try to add to it. In order to keep communication exchanges coherent, they need to keep track of their CG as it accumulates, increment by increment. These strata are not static constructions cast in concrete. Elaborate CG constructions can be destroyed if new information destroys the suppositions that have grounded prior contributions.

P2P PEERGROUPS AND COMMUNITIES

Introduction to peergroups

The online equivalent in peer-to-peer (P2P) computing of a cultural community that is likely to share CG is the peergroup. Peergroups in P2P systems act as virtual social spaces in which peers can interact and exchange information. For instance, in the JXTA protocol, the technical definition of a peergroup is a collection of peers that have agreed upon a common set of services (Sun Microsystems, 2002). Peergroups are user-created and motivated: it is up to cooperating peers (i.e. ordinary users) to define, groups, and leave groups (Oaks, Traversat, & Gong, 2002). Peergroups tend to be bottom-up social constructions in which it is up to members themselves to define the group's membership requirements (if any). Within P2P communities certain social norms can emerge, based on the recurrent social interactions that take place within them (Cumming, 2005). These communities can also witness accumulation of CG into hierarchical arrangements, or strata, which are useful in informing collaborative activity. Typically, though, there is no formal representation of CG structure beyond that which is distributed in the minds of those who frequent these peergroups.

Cultural communities and hierarchical competencies

Cultural communities also witness accumulation of strata of CG. These strata form notions of the social norms appropriate within these communities. Within normal communication there is no formal representation of these CG strata, even though their construction and maintenance plays a central role.

The types of inferences one can make based on information about communities are an important aspect of CG. Clark proposes a hierarchical representation between cultural communities in which parent-child relations exist. For example in the space of academic communities, one child of that could be the psychological academic community. Membership in the general academic community could give one reasons to presuppose some facts and competencies common to all academics, while the psychology specialization could allow more specialized inferences to be made. These inferences could involve what members might know, and about what references they might make in their interactions. Knowledge of social norms in specific communities, combined with knowledge of who belongs to these communities, can be a valuable social resource if this knowledge is well grounded.

Within P2P peergroups, specialized services and access to particular types of content and social processes are available, similar to that in other types of social communities. If one has knowledge of the specific norms within these peergroups this can give an outsider valuable information about where to find specialized services and information.

Processes of grounding in P2P communities

Clark's notion of the 'personal diary'

The CG model involves knowledge sources from both so-called communal sources and from personal experiences. Such knowledge enables people to make contributions to discourse, which are relevant to prior contributions and the accumulated CG within particular communities. Clark examines what sort of memory is required for people to maintain a CG as they interact with others. This personal information that records the interactions that people personally experience, Clark calls a 'personal diary' (Clark, 1996: 114). Personal diaries are taken from the perspective of only one person and record a log of events that have been personally experienced, for which the diarist has had direct personal involvement. They document events of a special type–joint perceptual experiences, or joint actions. That is, both parties must directly experience and focus on some entity, or participate in an event that requires their joint action, such as jointly participating in a conversation held during a design meeting. For example, CG management is required for designers to make relevant comments regarding the current state of a design. Important sources of information that can inform a design project are the data found within the personal diaries of all design participants.

Clark's notion of the personal diary and of the accumulation of CG does not assume any computational or other representational implementation. Instead, it provides a general cognitive model of knowledge creation derived from collaborative interactions, such that inferences can be made from these interactions. The information contained in this memory is of a personal kind that derives from personally experienced conversations. Contents of a personal diary cannot be readily shared, in their entirety, with others. Because of both privacy concerns and the impracticalities of sharing the density of information that is theoretically collected in such diaries, most would choose to share or disclose only a small proportion of their contents–even if it were technical feasible. Therefore, a personal diary is not recorded in any community-relevant document to which a whole community might refer, such as an encyclopaedia or a book of cultural norms.

Processes of personal diary management

This necessity of personal diary to inform collaborative processes implies certain processes: 1. *Recording interactions*: recording all people that an interlocutor has interacts with, within a specific process. For instance, for the purposes of recording a design process, all the interactions that a designer has with others should be recorded as far as they concern the design process: for example, all phone calls, design meetings, and informal chats by the drawing board. Note that these are must be joint actions that involve more than one person. 2. *Recording shared objects of attention within these interactions*: all entities that interlocutors have chosen to focus on such that they become objects of joint salience and recognized as such by all parties. 3. *Recording what is agreement upon concerning these objects*: whether to create new ones, or to modify or delete existing ones.

Personal diaries in P2P systems

In P2P systems, data also accumulate in a type of personal diary. In P2P systems this personal diary is called the local data cache. Within this local cache, the peer holds information that informs it about the state of the external environment and the peer's relationship to this external world. In P2P systems, a similar situation to that of signalling in speech acts exists, in which the goal is to give the impression that a group of people share certain perceptions and have access to common information resources. However, the means to do this are distributed between autonomous agents. In P2P systems, the signals communicated are in the form of text, often as XML-encoded messages. In the JXTA framework, these messages are called advertisements. A peer in a P2P system must keep track of these messages, since their content is what gives agents knowledge of their local context and they are what connect the P2P community together. Therefore, they are seen as a direct analogue of what in structuration theory is called social 'structure'. In this paper, a system of CG strata construction and management is described, which involves a hierarchical representation of these advertisements.

IMPLEMENTATION

Approach

CG is viewed as the social structure or cohesiveness that accumulates when people work together in teams: Within design teams that work with one another, social structure or social cohesiveness is incrementally increased as team members work together. This structure consists of accumulations of CG. Therefore, the presence of CG is an indication that a social group has created bonds between those on the team, while its absence implies that the team is lacks social cohesion. Social cohesion or solidarity, which deals with how social collectives are bound together, is a key issue in the sociological study of groups. According to Moody and White, study of such cohesion can be partitioned into an *ideational* component, referring to the psychological identification of members within a group and a *relational* component, referring to the connections among members of the group (Moody & White, 2001: 3). In the approach here, all CG derives from explicit, relational connections enabled by P2P communication, rather than through less-formalized psychological identification.

All grounding occurs via P2P-facilitated interaction: All grounding takes place in interpersonal interactions mediated using a P2P application, which is enabled through communication of XML-encoded messages between peers. All grounding takes place in online P2P interactions between actual peers, working within peergroups. Peergroups are virtual social spaces that serve as virtual places to put data within P2P systems. All information exchanged between peers gets placed in only peergroup at a time. If communication exchanges take place outside of the P2P system, within such a model, they cannot contribute to the CG.

Communication that is intended to be grounded consists of simple verbal content: All information exchange and reference takes place using simple words, rather than more complex media such as design representations of geometric entities, or other design documents. Objects of interest are called ‘groundable items’. These are represented as simple strings of indefinite length. There is no other formalization of what these groundable items can be, and this is up to the peers to define this for themselves. Examples could include: ‘the column near the front entrance’, or ‘floor finish in the library’. This approach does not involve the fine-grained propositions or discourse elements to found in linguistics or conversational analysis (Traum, 1994).

Basic levels of evidence: the rules that determine that agents are referring to the same object are intended to be very simple. These include such behaviours as: explicit acknowledgement, and continued interaction. That is, if an object is explicitly acknowledged, then it is considered to be grounded; also, if a person continues to interact with respect to an item of mutual interest, then that item is considered to be grounded and then can be considered as part of the CG. Peers are assumed to be interacting when they exchange messages. This need not be synchronous communication, but can also occur with indefinite delays. Normally co-presence helps the grounding process considerably, since peers have access to non-verbal clues and other contextual information. Here, co-presence is assumed simply if people continue to interact using the P2P application.

Automation of CG inference and emergence: All mechanisms to establish whether an item is part of the CG or not, do not require complex decisions or assessments by the user (or peer). Rather, the approach is to automate such a process such all such inference is performed automatically by the application. Such automation is intended so that CG management does not add to the already considerable cognitive burdens on designers within collaborative design processes. The three things inferred by the system are: 1. *Interactions*: this establishes whether peers not only send messages to one another, but whether they interact. Therefore, the receipt of a message requires that the recipient peer responds within a reasonable time. 2. *Groundable items*: these are the simple verbal elements that are discussed within a discourse. The simple fact that a groundable item is discussed within the context of an interaction implies that it is grounded. 3. *Peergroups*: peergroups are normally intentional constructions within P2P systems. That is, users intentionally create a new peergroup in which to communicate and interact. The type of content that is communicated within a particular may or may not have any commonalities and there may be no connection with the intended subject of the peergroup (assuming there is one). Having emergent peergroups means that a peergroup is by definition a place to discuss specific content because the system decides into which peergroup a groundable item is best placed. The inference mechanisms are detailed in Table 1, and the relations between user-created and application-inferred entities are shown in Figure 1.

Table 1: Inference mechanisms for emergent entities.

Inferred entities	**Inputs needed**	**Inference mechanism**	**Comments**
Interactions	Peers, messages	If peers respond to messages, that is, if they create conversational threads or perform conversational turn taking, then an interaction is created.	Peers responding to messages implies an interaction between them.
Common ground	Interactions, groundable items	If groundable items are discussed within interactions, then they become grounded and are added to the CG.	Continued interaction implies incrementing the CG.
Peergroups	Common ground, peers	If a set of peers creates CG together, then they form a community that can be represented by a peergroup.	Creating common ground together implies an online community or peergroup.

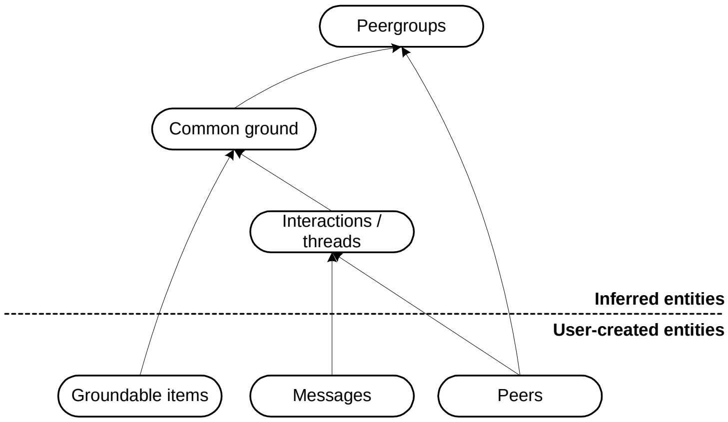


Figure 1: User-created and inferred entities.

Discussion

Grounding processes are seen to be essential for functional communication, however, it is unclear how to do this simply and in a way that doesn't burden users. Making simple inferences, based on the content of user caches and on the behaviours of peers using P2P systems, such as that described in (Cumming, 2005), appears to be a promising approach to the problem. One of the basic ideas of this approach is that communities, represented by peergroups, can be emergent entities that are formed as a result of the common ground created by users. Therefore, the type and nature of the content that is discussed between people is what determines what communities they belong to, rather than the other way around. This is in contrast to most P2P systems in which peergroups are simply intentional, non-emergent constructions that are useful in propagating information between a defined subset of peers.

This approach has the implication that communities can form and disappear quite quickly as grounded content fades within the time-dependent caches of P2P systems. P2P systems can be configured to such that their content can fade rather quickly. One option that negatively affects the basic functionality of a P2P system is when its content is configured to never fade – that is, when it has an infinite lifespan. Technically this is accomplished by simply setting the TTL (time-to-live) variable of the communicated messages (called advertisements in JXTA) to their maximum value. When content never fades in a P2P system, it can be inundated by obsolete content.

One aspect that requires further study and testing is what kind or rules are appropriate for the definition of common ground. It is unclear whether the approach of simply identifying that content which experiences continued interaction between peers is a strong enough basis to assume that this content should be considered part of the common ground. Perhaps stricter qualifications may be needed.

CONCLUSION

CG in normal language processes involves the distributed construction of grounded knowledge structures. CG provides a local context for communication informed by the direct personal experiences of people as they interact with others, and by expectations and norms of the various communities to which interlocutors belong. Grounding and management of accumulated CG is seen as a necessary sense-making process that enables interactive communication. CG management is required in all collaborative activities such as collaborative design, in which accumulation of CG is also necessary for the construction of coherent communication. Such knowledge enables the coordination of joint activity and communication of content of joint interest. Given the importance that CG has in coordination of action, it is also seen as essential for design coordination. Unfortunately there are few formal ways of accumulating CG in computer-mediated communication.

CG accumulation is an inherently distributed phenomenon. In whatever media, a process of grounding is required because people who interact do not have access to a communally shared database of the referential meaning of words, gestures, and social practices. This lack of a central representation requires the complex signalling and confirmative interactions between interlocutors that CG theory describes. This distribution of a communication and knowledge transfer process means that similar processes that support grounding in language processes can also inform distributed processes used in P2P systems. Grounding of items under discussion within online communities is essential for the management of personal and group communication. Keeping track of what is and isn't grounded is the essential task that peers must perform in order to continue to make sense of interactive communication. A simple approach designed not to increase cognitive burdens on users is described and involves the emergence of interactions, common ground, and peergroups within a P2P design coordination application.

REFERENCES

1. Andriessen, J. H. E. 2002. *Working with Groupware: Understanding and Evaluating Collaboration Technology*. London: Springer.
2. Birnholz, J., Finholt, T., Horn, D., & Bae, S. J. 2005. Grounding Needs: Achieving Common Ground Via Lightweight Chat In Large, Distributed, Ad-Hoc Groups. *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. Portland, OR. 21-30.
3. Blau, J. 1984. *Architects and firms: a sociological perspective on architectural practice*. Cambridge, MA: The MIT Press.
4. Bowker, G., & Star, S. L. 1999. *Sorting Things Out: Classification and its Consequences*. Cambridge, MA: The MIT Press.
5. Bucciarelli, L. L. 1994. *Designing Engineers*. Cambridge, MA: The MIT Press.
6. Bucciarelli, L. L. 2003. *Engineering Philosophy*. Delft, Netherlands: DUP Satellite, Delft University Press.
7. Clark, H., & Brennan, S. 1991. Grounding in Communication. In L. Resnick & J. Levine & S. Teasley Eds. *Perspectives on Socially Shared Cognition*. 127-149. Washington, DC: American Psychological Association.
8. Clark, H. H. 1996. *Using Language*. Cambridge, UK: Cambridge University Press.
9. Cumming, M. 2005. *Constructing Process Models from Distributed Design Activity*. Ph.D. thesis. School of Architecture, Carnegie Mellon University, Pittsburgh.
10. Fischer, G. 2004. Social creativity: turning barriers into opportunities for collaborative design. *Proceedings of Eighth Conference on Participatory Design: Artful Integration: Interweaving Media, Materials and Practices*. Toronto, Ontario, Canada. 152-161.
11. Flores, F., Graves, M., Hartfield, B., & Winograd, T. 1992. Computer Systems and the Design of Organizational Interaction. In D. Marca & G. Bock Eds. *Groupware: Software for Computer-Supported Cooperative Work*. Los Alamitos, CA: IEEE Computer Society Press.
12. Geisler, C., & Rogers, E. H. 2000. Technological mediation for design collaboration. *Proceedings of IEEE Professional Communication Conference* and *Proceedings of the 18th annual ACM Conference on Computer Documentation: Technology and Teamwork*. 395-405.
13. Giboin, A. 1998. The use of Herbert H. Clark's models of language use for the design of cooperative systems. *SIGGROUP Bulletin*, ACM Press. 19. 2. 27-31.
14. Gutman, R. 1988. *Architectural practice: a critical view*. Princeton, NJ: Princeton Architectural Press.
15. Hendry, D. 2004. Communication functions and the adaptation of design representations in interdisciplinary teams. *Proceedings of Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. Cambridge, MA, USA. 123-132.
16. Hoadley, C. M., & Kilner, P. G. 2005. Using technology to transform communities of practice into knowledge-building communities. *SIGGROUP Bulletin*, ACM Press. 31-40.
17. Larsson, A. 2003. Making sense of collaboration: the challenge of thinking together in global design teams. *Proceedings of ACM SIGGROUP Conference on Supporting Group Work*. Sanibel Island, Florida, USA. 153-160.
18. Moody, J., & White, D. 2001. Structural Cohesion and Embeddedness: A Hierarchical Conception of Social Groups. *Working Paper 00-08-049*. Santa Fe, NM: Santa Fe Institute.
19. Nardi, B., & O'Day, V. 1999. *Information Ecologies: Using Technology with Heart*. Cambridge, MA: The MIT Press.
20. Nardi, B. A., Whittaker, S., & Bradner, E. 2000. Interaction and outeraction: instant messaging in action. *Proceedings of ACM Conference on Computer Supported Cooperative Work*. 79-88.
21. Oaks, S., Traversat, B., & Gong, L. 2002. *JXTA in a Nutshell*. Sebastopol, CA: O'Reilly.
22. Orlikowski, W., & Yates, J. 1998. Genre Systems: Structuring Interaction through Communicative Norms *Working Paper CCS WP #205, Sloan WP #4030*. Cambridge, MA: MIT Sloan School of Management.
23. Schelling, T. 1978. *Micromotives and Macrobehavior*. New York, NY: Norton.
24. Schön, D. 1983. *The Reflective Practitioner: How Professionals Think in Action*. New York, NY: Basic Books.
25. Star, S. L., & Ruhleder, K. 1994. Steps towards an ecology of infrastructure: complex problems in design and access for large-scale collaborative systems. *Proceedings of ACM Conference on Computer Supported Cooperative Work*. 253-264.
26. Sun Microsystems, I. 2002. *Project JXTA v2.0: Java Programmer's Guide*. Sun Microsystems Inc. Available: http://www.jxta.org/docs/jxtaprogguide_final.pdf [2003, 30 May].

27. Traum, D. 1994. A Computational Theory of Grounding in Natural Language Conversation. *Technical Report and Ph.D. Thesis ROC 545*. Rochester, NY: Computer Science Department, University of Rochester.
28. Traum, D., & Allen, J. 1994. Discourse obligations in dialogue processing. *Proceedings of 32nd annual meeting on Association for Computational Linguistics*. Las Cruces, New Mexico. 1-8.
29. Ure, J., Lloyd, A., Pooley, R., & Dewar, R. 2003. Scaffolding knowledge sharing and decision support in distributed web-based systems: a socio-technical perspective. *Proceedings of CE 2003 Concurrent Engineering: Research and Applications*. Madeira, Portugal. 26-30 July 2003. R. Jardim-Gonçalves & J. Cha & A. Steiger-Garção Eds. 1075-1082.
30. Walker, M. 1997. Review of "Using language" by Herbert H. Clark. Cambridge University Press 1996. *Computational Linguistics*. 23. 4. 625-628.
31. Wenger, E. 1999. *Communities of Practice: Learning, Meaning and Identity*. Cambridge, UK: Cambridge University Press.

Multidimensional Multilevel Networks in the Science of the Design of Complex Systems

Jeffrey Johnson

Department of Design and Innovation, The Open University
Milton Keynes, MK7 6AA, UK
j.h.johnson@open.ac.uk

ABSTRACT

Assembling parts under relations to form wholes is the *Fundamental Construct of Multilevel Systems*. Intermediate assemblies themselves become assemblies, defining multilevel structure. If the set of parts exists at one level, the set structured by an assembly relation exists at a higher level. The structured set with its assembly relation defines a multidimensional simplex at this higher level. A simplex is a multidimensional generalisation of a link in a network: a relation between three things being a triangle (two-dimensional simplex), a relation between four things being a tetrahedron (three-dimensional simplex), and so on. Sets of simplices have a multidimensional connectivity that can be analysed by *Q-analysis*, and a more refined method called *star-hub* analysis. Star-hub systems also have a Galois lattice structure. Connectivity between simplices acts as a kind of relatively static *backcloth* for more dynamic patterns of numbers called the *system traffic*. Relationships between numerical mappings constitute the *Order-I* dynamics of the system, while changes in the backcloth constitute *Order-II* dynamics in the system. Multidimensional connectivity constrains the horizontal intra-level Order-I dynamics and the Order-I inter-level dynamics. Order-II dynamics concern the *building* of structure and the *annihilation* of structure, and are discrete and non-linear. A theory of design is presented using this multidimensional multilevel network theory. Designers *build* structures in bottom-up and top-down fashion. Top-down involves hypothesising sets of parts and relationship to aggregate into higher level abstract constructs. Bottom-up involves assembling real things into realisable structures under explicit relations. As top-down meets bottom-up, abstractions are instantiated with tangibilities, and eventually the whole design becomes grounded in tangible things. This is the *blueprint* stage at which the design can be fabricated. To achieve the blueprint it is necessary to follow a dynamic creative process, the *design process*, which is sensitive to initial conditions, computationally irreducible, path-dependent and characterised by emergence and coevolution between the designed system and the requirements and specification of that system. This is a *science of the artificial*: if the designer creates a system that did not exist before, they are the first person to accumulate and synthesise knowledge about that system. Thus the designer acts as a scientist, by building the *representation* of the system, making *hypotheses* about the system within the *language* being constructed, performing *experiments* on the system, and synthesising this into a *theory* of the system and its dynamics. Many scientists interested in complex artificial systems are motivated by the possibility of using that scientific knowledge to manipulate the system, either by designing new systems, or modifying and managing the behaviour of existing systems. Thus not only are the designers of artificial systems scientists, but the *scientists of artificial systems are designers*. During the meetings of the *Embracing Complexity in Design* cluster it has become clear that designers across the disciplines share a culture based on the creation of new systems and the management of existing systems. In particular the *design process* is common to all design domains, from graphic design through architecture through software to engineering design. This culture informs the particular design process, supporting creativity and divergence, and leading to convergence and delivery of results. It is suggested that scientists of the artificial would benefit from accepting that they are acting as designers, and that complexity science has much to learn from the design community.

Keywords: multilevel, multidimensional, systems, hierarchy, networks, simplicial complexes, Q-analysis, backcloth, traffic, design, artificial systems, synthetic systems, complex systems.

1. INTRODUCTION

This paper will develop the argument that design is at the heart of creating a science of complex systems. Let us make a distinction between natural systems and artificial systems: *natural systems* will be those that exist without human intervention; *artificial systems* will be those that exist as a result of accidental or deliberate human intervention. Natural systems are studied as they are. Artificial systems can be physical objects such as the products one buys, they can be human systems such as a choir, and they can be socio-technical systems such as cities, universities, armies, and businesses which are made up of physical and human parts. Artificial systems can be studied as they are with no intention of introducing change, as in anthropology, but more often they are studied in the context of what they *ought* to be[1]. Disciplines such as anthropology, psychology, and sociology can be free of judgements of what human systems ought to be, but often they feed into policy inducing change. Disciplines such as city planning and peace studies are explicitly linked to policy and managing socio-technical dynamics. Systems that do not already exist are created by *design*.

The designer of a completely new system is the first person to know anything about that system. The design process involves collecting information as the system is created, and the designer is the first person to bring that knowledge together into a *theory* of the new system. *Designers are the first scientists to investigate the systems they create.*

Although the motivation in studying complex systems may involve pure scientific curiosity, in many cases it includes the desire to change systems and system behaviour. In other words, for artificial systems *science proceeds through design*. There are many examples: the biochemist *designs* the molecule; the physician *designs* the treatment; the roboticist *designs* the robot; the planner *designs* the city; the administrator *designs* the organisation; and so on.

At the heart of design is the idea that one can take the 'right' set of parts and put them together the 'right' way to produce a system with certain pre-specified properties[2]. As will be seen, finding the right parts and the right way to put them together is a non-trivial process, involving the construction of an explicit *description* of the system and the accumulation of knowledge about the system.

The design process involves clarifying requirements and specifications, using various methods to find the 'right' set of parts and the 'right' way to put them together to satisfy the specification. At the end of the process there is a *blueprint*, detailing all the parts and the way they must be put together.

A scientist can make three kinds of observation:

(1) observing that something *exists*

(2) counting and assigning *numbers* to things

(3) observing *relationships* between things

Traditional science has focused on the first two of these, but increasing it is realised that *relationships* are important in complex systems. For many years relationships in human systems have been studied using network theory, and it is now clear that network properties play a fundamental role in the behaviour of complex systems[3]. Mathematical relations play a fundamental role in this paper, as a means of making the multilevel nature of systems well defined, and defining the multidimensional spaces on which patterns of numbers can be defined. The dynamics of systems can be played out both through changes in the numbers (Order-I dynamics) or changes in the multidimensional structure (Order-II dynamics)[4].

In design one often *observes in the mind's eye* things that do not exist, *i.e.* designers envisage things that *could* exist but do not yet exist. But designers do not do this instantaneously: *e.g.* aeronautical engineers do not suddenly find fully instantiated designs for a new aeroplane in their heads, architects do not suddenly find fully instantiated designs for a building in their heads, and electrical engineers do not suddenly find fully instantiated designs for circuit boards in their heads. These designs emerge from a process – the *design process*.

In this paper I will sketch a theory of how the creative design process works. For complex systems it involves building a language to represent that system. Often this involves creating new parts and giving them names, and assembling them to create a multi-level ensemble. Part of the process is bottom-up – putting together existing things under new or known relationships. Another fundamental part of the process is top-down, replacing high level uninstantiated abstractions with more detailed abstractions at a lower level. Eventually the bottom-up and top-down meet, and the design becomes a fully instantiated blueprint, with every construct at every level *grounded* in elements that exist in some observable way. To achieve this involves a lot of reasoning about how the parts might interact and what their emergent properties might be. Occasionally higher level reasoning in abstractions leads to errors. As the design is instantiated at lower levels things that are expected to fit fail to do so. Such design problems may be solved by ingenuity, but when they cannot, the design has to back-up with higher level assumptions being modified. Generally this is highly undesirable, implying loss of time, disruption to schedule, and related expense.

In the context of this theory of design, I will claim that the science of artificial systems follows the path of the designer. It is clear that designers are the first scientists to build theories of the artificial systems they create. I will argue that scientists investigating artificial systems with a view to applying that knowledge *are* designers, whether they know it or not. In other words, the design process *is* the scientific process, and the systematic body of knowledge known as 'design research' can be extremely valuable to those engaged in the science of artificial systems.

2. MULTILEVEL SYSTEMS

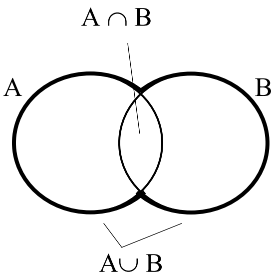


Figure 1. Euler circles representing sets and set operations

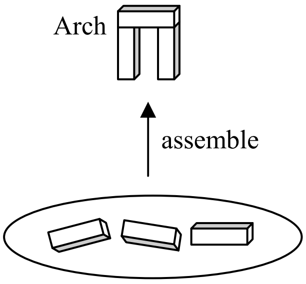


Figure 2. Assembling a set of parts to form a whole

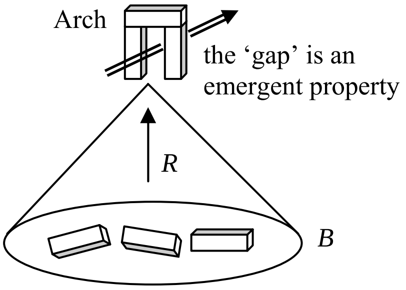


Figure 3. A hierarchical cone

Figure 1 shows the use of *Euler circles* to represent sets and their intersection and union. These diagrams are used extensively in set theory, and we use them to represent sets of parts. In our diagrams the Euler circles will usually be draw in perspective, and so appear as *Euler ellipses*. Figure 2 uses this idea to show how the elements of a set of a set of blocks can be assembled to form a structured object called an *Arch*. The set of parts is represented by the blocks enclosed in an Euler ellipse. These are then assembled to form a structure represented by its *name*, here Arch. Using the convention that higher level objects appear higher on the page than lower level objects, one can draw a *hierarchical cone*, as illustrated in Figure 3, showing how the parts aggregate into the structure under the *assembly relation*, R. If B is the set of blocks, we can write $R(B)$ as the result of applying R to B, so that $R(B)$ = Arch.

Figure 4 shows the *Fundamental Construction of Multilevel Systems* in which wholes are assembled from sets of parts under an *assembly relation*. The whole may have properties not possessed by its parts, *e.g.* the arch has a 'gap' between its vertical support blocks. These 'emerge' from the construction and are called *emergent properties*. The arch is a structured set, denoted $\langle b_1, b_2, b_3 ; R \rangle$. We use the convention that an object and its name can be used interchangeably, so $R(B) = \langle b_1, b_2, b_3 ; R \rangle = Arch$.

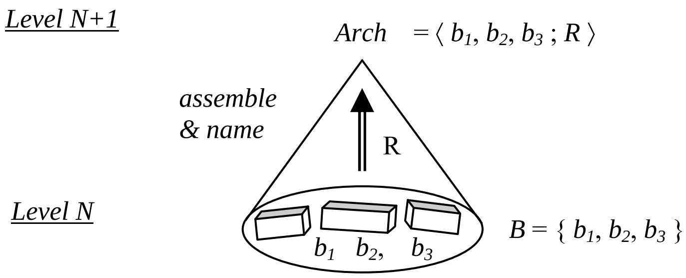


Figure 4. The Fundamental Construction of Multilevel Systems

In general systems have many levels, as illustrated in Figure 5. Here *lines* at the lowest atomic level, which is denoted *Level N*, are assembled under the relations R_1, R_2, and R_3 to form blocks of various kinds at a *higher hierarchical level*, denoted *Level N+1*. These blocks are assembled by relations R_4, R_5, and R_6 to form shapes of various kinds at another higher level, denoted *Level N+2*. Finally these three shapes are assembled by the relation R_7 into the final shape at what is called *Level N+3*.

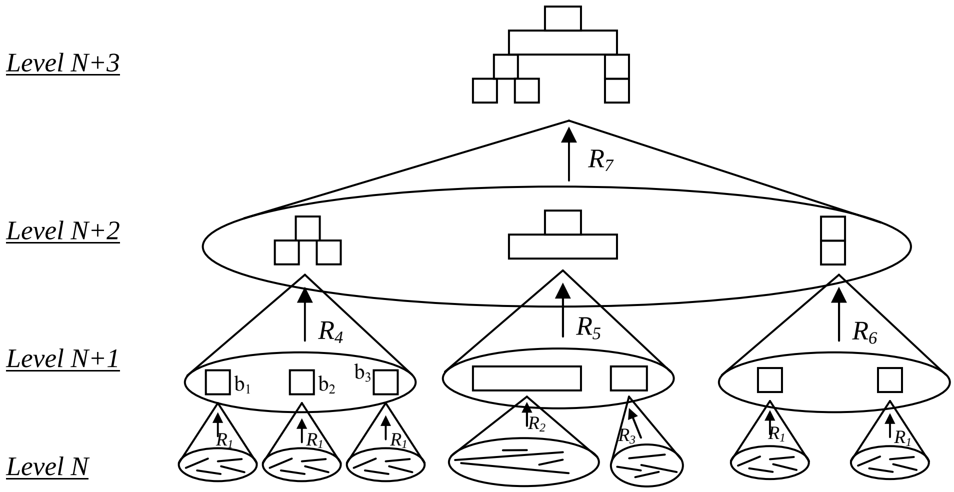


Figure 5. Hierarchical aggregation in multilevel systems

The notation $N+k$ is used to show that the levels are not absolute. In general there are many ways to aggregate through the hierarchy of assembly, and it common to introduce a new level between two existing levels to represent intermediate assemblies. In design there may be many ways to put things together, but usually just one of them is selected for the final blueprint.

3. RELATIONAL STRUCTURE

R	a_1	a_2	a_3	a_4
a_1	0	1	1	0
a_2	0	0	1	1
a_3	1	0	0	0
a_4	0	0	0	0

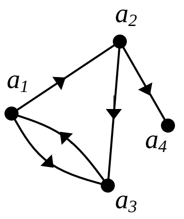


Figure 6. The graph of a relation from a set to itself

R	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_1	1	1	1	0	0	0	0	0
a_2	0	1	1	1	0	0	0	0
a_3	0	0	1	1	1	1	0	0
a_4	0	0	0	0	0	1	1	1

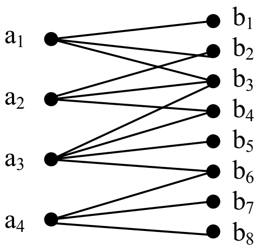


Figure 7. A bipartite graph between sets *A* and *B*

A *relation*, *R*, between two sets *A* and *B* is an operational rule for deciding of each *a* in *A* and each *b* in *B* whether or not *a* is related to *b*. A relation can be recorded as an *incidence matrix*, as illustrated in Figures 6 and 7. In Figure 6 the relation is between a set and itself, while in Figure 7 it is between two different sets. Next to each matrix is a diagram showing the elements as points and the relationships by lines linking points. Technically, a *graph* is a set of *vertices*, *V*, and a set of pairs of elements of *V*, denoted $\langle v_1, v_2\rangle$, called *edges*. In Figure 6 the graph has the vertices $\{ a_1, a_2, a_3, a_4 \}$ and the edges $\{ \langle a_1, a_2\rangle, \langle a_1, a_3\rangle, \langle a_2, a_3\rangle,\langle a_2, a_4\rangle,\langle a_3, a_1\rangle \}$. When the edges are *directed* with $\langle v_1, v_2\rangle \neq \langle v_2, v_1\rangle$ they may be drawn as arrows, as shown in Figure 6. Such directed graphs are often called networks. Graphs and networks have many useful properties. For example, the *degree* of a vertex is the number of incident edges, and is a direct measure of how many things that vertex is related to. One of the most important features of networks is the *connectivity*, which allows things to flow through the system, and can be responsible for one part of the system influencing another.

In general relations are defined between different sets, as illustrated in Figure 7. This gives rise to what is called a *bipartite graph*. In Figure 7 the vertices are { a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8 } and the edges are { $\langle a_1, b_1\rangle$, $\langle a_1, b_2\rangle$, $\langle a_1, b_3\rangle$, $\langle a_2, b_2\rangle$, $\langle a_2, b_3\rangle$, $\langle a_2, b_4\rangle$, $\langle a_3, b_3\rangle$, $\langle a_3, b_4\rangle$, $\langle a_3, b_5\rangle$, $\langle a_3, b_6\rangle$, $\langle a_4, b_6\rangle$, $\langle a_4, b_7\rangle$, $\langle a_4, b_8\rangle$ }.

The bipartite graph of a relation between two different sets may appear rather uninteresting, but the degrees of the vertices are related to a much richer structure, namely a *hypergraph*. Given a set *V* of vertices, a *hypergraph* is that set of vertices together with a class of subsets of the vertices, called *hyper-edges*. For example, the relation in Figure 7 defines a class of subsets of *B* given by { $a_1 \rightarrow \{b_1, b_2, b_3\}$, $a_2 \rightarrow \{b_2, b_3, b_4\}$, $a_3 \rightarrow \{ b_3, b_4, b_5, b_6\}$, $a_4 \rightarrow \{b_6, b_7, b_8\}$ }, which can be considered to be a hypergraph. A more interesting hypergraph also includes the intersections of the sets, as shown in Figure 8. Let this hypergraph be denoted $\mathrm{H_A}(B, R)$. This hypergraph is defined by the elements of *A* being related to subsets of *B*. The *conjugate hypergraph*, $\mathrm{H_B}(A, R)$, is defined by the elements of *B* being related to subsets of *A*, as shown in Figure 9.

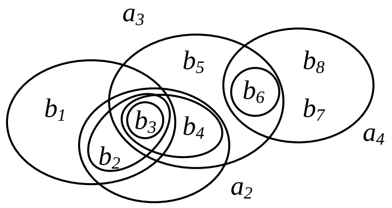


Figure 8. The hypergraph $\mathrm{H_A}(B, R)$

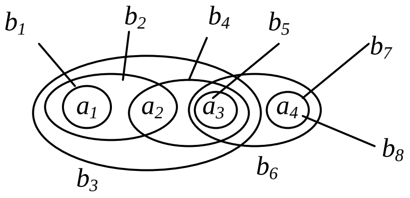


Figure 9. The hypergraph $\mathrm{H_B}(A, R)$

These two hypergraphs have an intimate interrelationship. Every hyper-edge of $\mathrm{H_A}(B, R)$ is associated with a hyper-edge in $\mathrm{H_B}(A, R)$, with all elements related. For example { a_1, a_2 } is associated with $\{b_2, b_3\}$, with both of a_1 and a_2 related to both of b_1 and b_2. These pairs of sets can be arranged in what is called a *Galois lattice*, illustrated in Figure 10. Here ∅ is the empty set, and the expressions ({ a_1, a_2, a_3, a_4 }, ∅) means that no member of *B* is related to every member of *A*, while and (∅, $\{b_3, b_6, b_7, b_8\}$) means that no member of *A* is related to every member of *B*.

A *lattice* is a partially ordered set is which every two elements have a supremum and an infimum. In Figure 10 we say that $(x, y) < (x', y')$ if $x \subset x'$ and $y \supset y'$. For any (x, y) and (x', y'), $(x \cup x', y \cap y')$ exists by construction. It is the supremum of (x, y) and (x', y'), since $(x, y) < (x \cup x', y \cap y')$ and $(x', y') < (x \cup x', y \cap y')$. Similarly $(x \cap x', y \cup y')$ belongs to the system, and is the infimum of (x, y) and (x', y'), since $(x \cap x', y \cup y') < (x, y)$ and $(x \cap x', y \cup y') < (x', y')$. In Figure 9 a line is drawn between each pair of sets and their supremum and infimum, to produce the lattice.

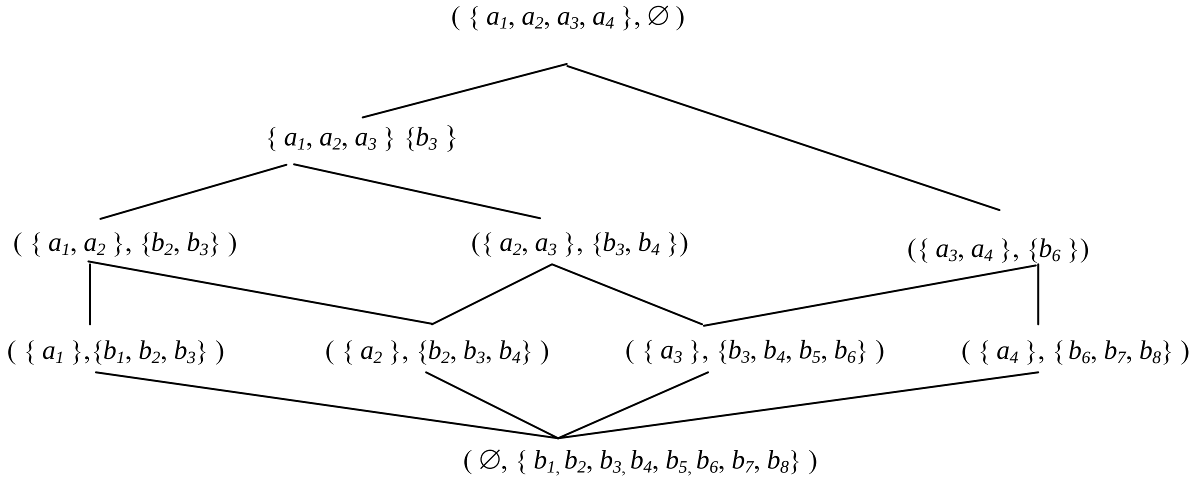
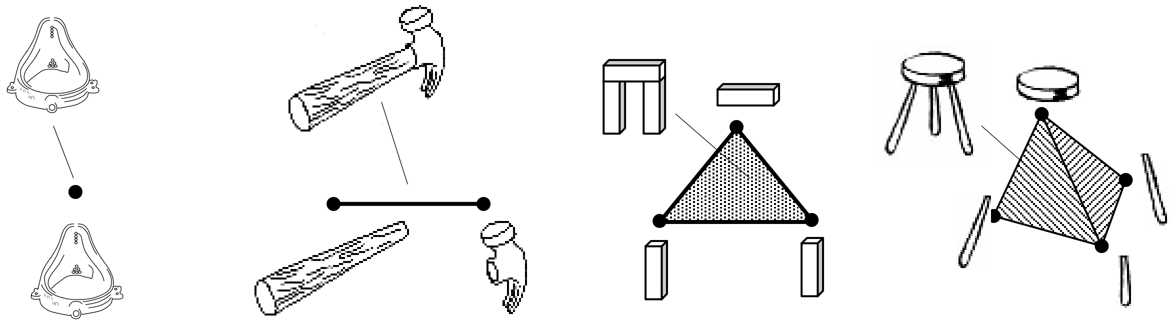


Figure 10. The Galois lattice of the relation *R* between *A* and *B*.

4. MULTIDIMENSIONAL NETWORKS AND Q-ANALYSIS

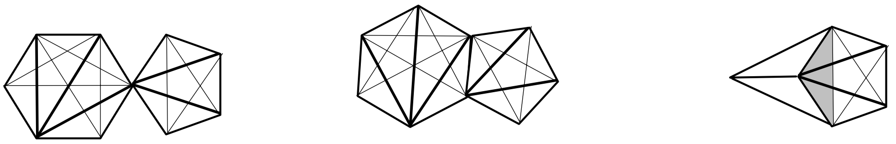
The Galois lattice construction sketched in the previous section is a powerful way of investigating relations in systems. However it is essentially *set theoretic*. In the light of Section 2, the underlying hypergraphs can represent the *sets* of components in multilevel systems, but they cannot also represent the *structured sets of components*. For this we need to make the assembly relations explicit, and develop the algebraic properties of the relations as a natural multidimensional generalisation of network theory.



(a) vertex or 0-simplex (b) line or 1-simplex (c) triangle or 2-simplex (d) tetrahedron or 3-simplex

Figure 11. Multidimensional simplices as the generalisation of links in a network

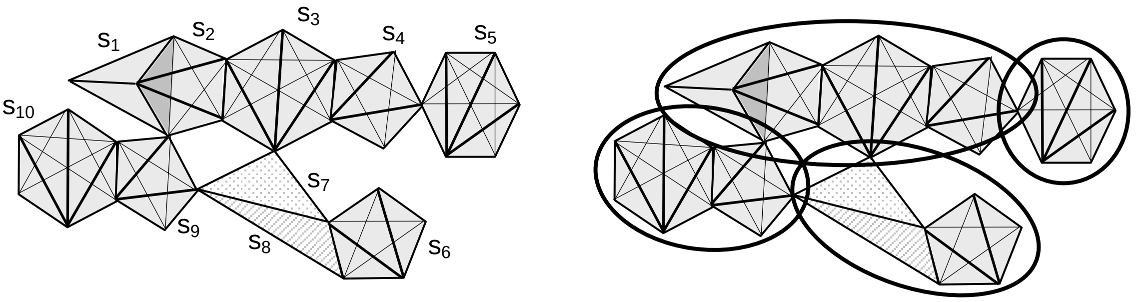
In Section 2 we introduced the notation $\langle b_1, b_2, b_3 ; R \rangle$ to represent a set of three blocks assembled into an arch. This is clearly more that the set $\{ b_1, b_2, b_3 \}$ which we drew as an Euler circle. In Figure 11(c) we show $\langle b_1, b_2, b_3 ; R \rangle$ drawn as a *triangle*, a two-dimensional object. This is the natural generalisation of using a one-dimensional *line* to represent a relation between two things, and a zero-dimensional point, or vertex, to represent a single thing. The structure $\langle b_1, b_2, b_3 ; R \rangle$ is called a *simplex* and it has *dimension* two. Figure 11(d) shows an object made up of four parts, being represented by a three-dimensional tetrahedron. In general a relation between *n* vertices is represented by an (*n*-1)-dimensional polyhedron in a multidimensional space. A simplex is a *face* of another if its vertex set is a subset of that simplex. A set of simplices with all its faces is called a *simplicial complex*.



(a) 1 shared vertex (b) 2 shared vertices (c) 3 shared vertices

Figure 12. Simplices can be connected at different dimensions

Simplices have interesting connectivity properties. Figure 12 shows how simplices can share different numbers of vertices, and that the more vertices they share, the more *highly connected* they are. The intersection of two simplices is called their *shared face*. If the shared face has dimension *q*, the simplices are said to b *q-near*. Thus the simplices in Figure 12(a) are 0-near (a single vertex has dimension zero), those in Figure 12(b) are 1-near (two vertices make a one-dimensional line), and those in Figure 12(c) are 2-near (three vertices make a two-dimensional triangle). We say two simplices are *q-connected* if there is a chain of pairwise *q-near* simplices between them.



(a) the simplicial complex

(b) components connected by lines

Figure 13. A simplicial complex of connected simplices[4].

A *Q-analysis* of a simplicial complex is a listing of its *q-connected components*. For example, for Figure 13(a):

$q = 5$ $\{ s_3\}, \{s_5\}, \{s_{10}\}$
$q= 4$ $\{s_2\}, \{s_3\}, \{s_4\}, \{s_5\}, \{s_6\}, \{s_9\},\{s_{10}\}$
$q=3$ $\{s_1\}, \{s_2\}, \{s_3\}, \{s_4\}, \{s_5\}, \{s_6\}, \{s_9\},\{s_{10}\}$
$q = 2$ $\{s_1, s_2\}, \{s_3\}, \{s_4\}, \{s_5\}, \{s_6\}, \{s_7\}, \{s_8\}, \{s_9\},\{s_{10}\}$
$q = 1$ $\{s_1, s_2, s_3, s_4\}, \{s_5\}, \{s_6, s_7, s_8\}, \{s_9, s_{10}\}$ (illustrated in Figure 13(b))
$q = 0$ $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$

Figure 14 shows the relationship between four English public houses and the customers that frequent them. Typically people like to go to more than one pub on different days for the variety it brings. Suppose that someone who likes The Swan, the simplex on the left, knows a very good joke. When he gets to the Swan pub he tells it to the people who happen to be in that day. They may tell the joke to other people in the pub, and it is likely to be transmitted to everyone in the Swan before the day is finished. The next day, one of those people in the Swan might visit the Anchor, and tell the joke there. Again the joke gets transmitted within the pub. The next day one of the people from the Anchor might visit the Goat pub, and tell the story there. In this way the joke can get transmitted from the Swan pub to the Bull pub, even though they have no customers in common. This illustrates how information can pass through social structure determined by relations. In general the more highly connected the structure, the more rapidly information is transmitted[4].

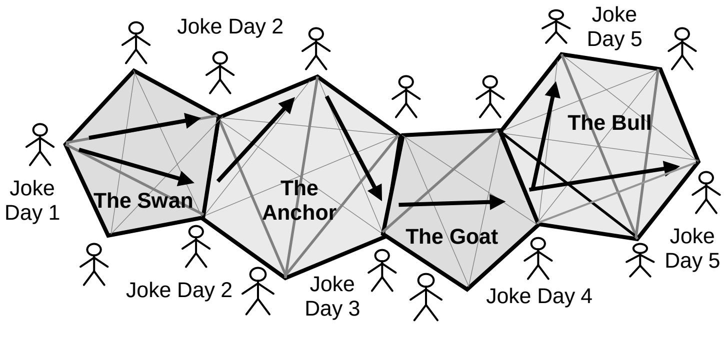


Figure 14. The transmission of information on the pub-customer structure[4]

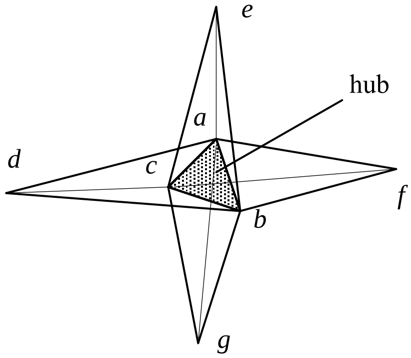


Figure 15. A star-hub configuration

Q-analysis is based on the connectivity of pairs of simplices, but it is more general to consider the intersection of sets of simplices. This leads to the concept of stars and hubs, as illustrated in Figure 15 where the simplices, $\langle a, b, c, d \rangle$, $\langle a, b, c, e \rangle$, $\langle a, b, c, f \rangle$, and $\langle a, b, c, g \rangle$ share the face $\langle a, b, c\rangle$. The set of the four simplices is called a *star* and their intersection is called their *hub*.

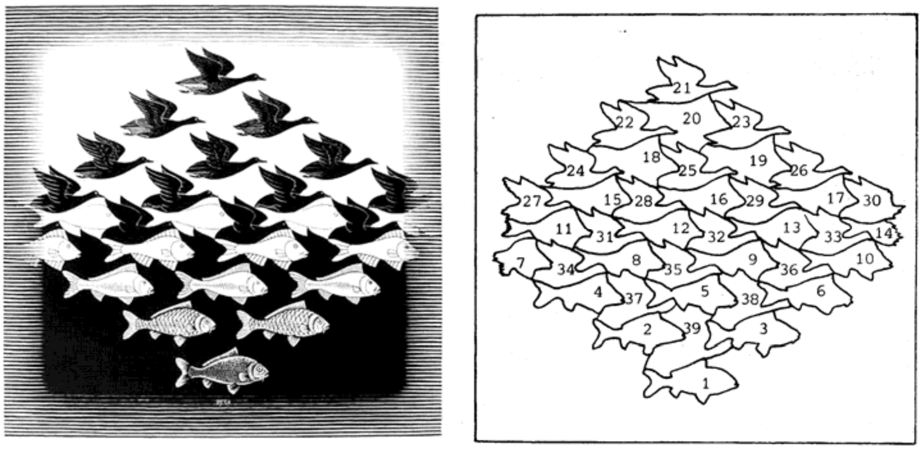


Figure 16. An analysis of Escher's Sky and Water[5]

If the rows and columns of an incidence matrix are arranged appropriately, hubs correspond to rectangular regions in which the values are all 1. For example, Figure 16shows thirty nine shapes abstracted from Escher's Sky and Water. The relation between these and the set of visual features {scales, mouth, gills, fish-tail, fins, fish-shape, eye, duck-shape, two-wings, feathers, beak, legs} is given by the incidence matrix below.

	1	2	3	4	5	6	8	9	10	11	12	13	7	21	22	23	24	25	26	28	29	31	32	33	27	30	34	35	36	37	38	14	15	16	17	18	19	20	39
scales	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mouth	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gills	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fish-tail	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
fins	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fish-shape	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
eye	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
duck-shape	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0
two-wings	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
feathers	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
beak	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
legs	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The matrix shows that the simplices for the shapes numbered 1 to 6 are all the same, with a block of 1s corresponding to the features { scales, mouth, gills, fish-tail, fins, fish-shape, eye },with all the shapes being good examples of fish. There is another group, shapes 1 to 13 all related to the hub features {fins, fish-shape, eye}, which includes the less perfect fish shapes 8 to 13. The following major star-hub pairs can be abstracted as follows:

$$\langle 1, 2, 3, 4, 5, 6 \rangle \leftrightarrow \langle \text{scales, mouth, gills, fish-tail, fins, fish-shape, eye} \rangle$$

$$\langle 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13 \rangle \leftrightarrow \langle \text{fish-tail, fins, fish-shape, eye} \rangle$$

$$\langle 21, 22, 23, 24, 25, 26, 28, 29 \rangle \leftrightarrow \langle \text{eye, duck shape, two wings, feathers, beak, legs} \rangle$$

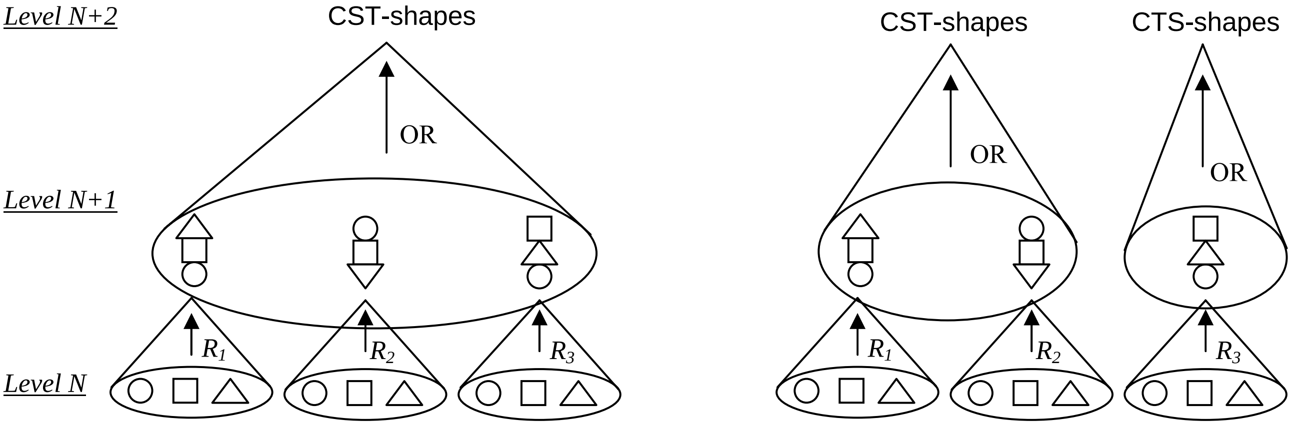
$$\langle 21, 22, 23, 24, 25, 26, 28, 29, 31, 32, 33, 27 \rangle \leftrightarrow \langle \text{eye, duck shape, two wings} \rangle$$

The final weak structure is given by the shapes 15-19 which are related only to fish-shape, and 34-38 which are only related to duck shape. Thus Escher has the fish shapes highly connected at the bottom of the picture and the bird shapes highly connected at the top. Moving towards the centre the connectivity is still high but less, and it gets relatively low as the shapes lose features and become the background.

5. OR - AGGREGATION IN MULTILEVEL SYSTEMS

The assembly relations discussed in the previous sections required *all* the elements to be present for the relation to hold between them, and introduced other conditions too. For example, in Figure 4 we require b_1 AND b_2 AND b_3 for the assembly into an arch. We call this kind of hierarchical aggregation an *AND aggregation*. This is different to another kind of aggregation, the *OR aggregation*, in which just one element is sufficient to move up the hierarchy.

For example, consider the Circle, Square and Triangle shapes in Figure 17. Let them be denoted by C, S, and T respectively. These have been assembled in three different ways by the relations R_1, R_2, and R_3. Are the structures the same, *i.e.* is $\langle C, S, T ; R_1 \rangle = \langle C, S, T ; R_2 \rangle = \langle C, S, T ; R_3 \rangle$? In some obvious sense they are not all equal because they are all different. However, for the purpose in hand it may not matter the centre shape an upside-down version of the leftmost shape. Indeed it may not matter that the rightmost shape has the blocks arranged vertically in a different order. In other words, for the particular purpose in hand all these shapes might be considered *equivalent*, each being an example of what might be called a CST-shape.



(a) any one of the shapes is sufficient for OR-aggregation (b) OR-classes are usually defined by purpose

Figure 17. Assembling shapes under an OR aggregation

For another purpose it may indeed matter that the blocks are assembled in a different order, leading to two classes: the CST-shapes = $\{ \langle C, S, T ; R_1 \rangle , \langle C, S, T ; R_2 \rangle \}$ and the CTS-shapes = $\{ \langle C, S, T ; R_3 \rangle \}$. This illustrates the general point that classifications are usually motivated by some purpose. Generally OR aggregations are a matter of definition to achieve a particular purpose.

There are two ways to define a class. The first is by *extension* or by listing the elements. For example, let *X* be the set {a, e, i, o, u}. Then some people call this the set of vowels. Others argue that *X'* = {a, e, i, o, u, y} should be called the set of vowels. The important thing is that by listing the elements, both *X* and *X'* are well defined. Which of them should be called the set of vowels is a matter of definition by *intension*.

Definition by *intension* involves giving a defining property(s). For example, Wikipedia defines vowel as follows: "In phonetics, a vowel is a sound in spoken language that is characterized by an open configuration of the vocal tract where there is no build-up of air pressure above the glottis. This contrasts with consonants, which are characterized by a constriction or closure at one or more points along the vocal tract. The additional requirement is that vowels function as syllabic units: it is this criterion that helps distinguish vowels from approximants (in some languages approximants may be slightly more constricted or less intense)", where "Approximants are speech sounds that could be regarded as intermediate between vowels and typical consonants. In the articulation of approximants, articulatory organs produce a narrowing of the vocal tract, but leave enough space for air to flow without much audible turbulence. Approximants are therefore more open than fricatives. This class of sounds includes lateral approximants like [l], as in lip, and the so-called semivowels [y] and [w] in yes and well." (http://en.wikipedia.org/wiki/Vowels, accessed 30-10-05).

In contrast to Wikipedia, http://arapaho.nsuok.edu/~gieseb/4323/phonicsterminology.html (accessed 30-10-05) gives an extensional definition "The letters a, e, i, o, and u represent vowel sounds", and then muddies the water by adding "and the letters w and y take on the characteristics of vowels when they appear in the final position in a word of syllable. The letter y also has the characteristics of a vowel in the medial (middle) position in a word of syllable."

These definitions are summarized in Figure 18.

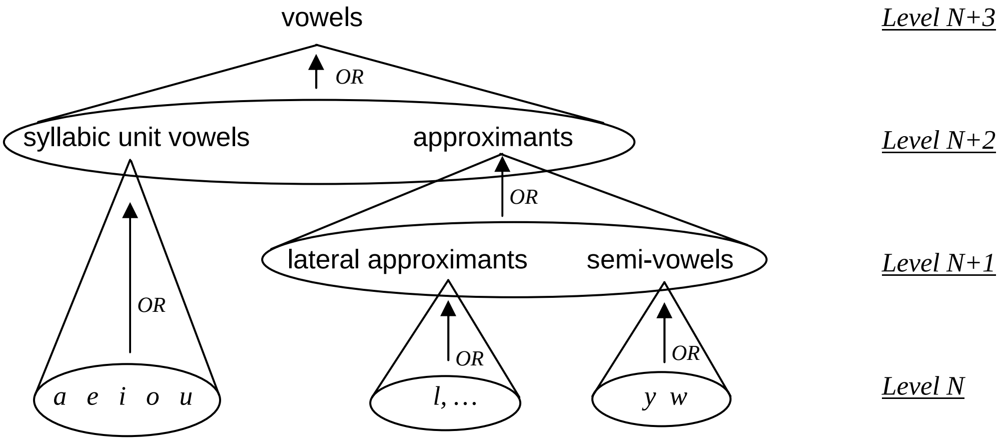


Figure 18. Higher level intensional definitions grounded in instantiated extensional definitions at lower levels.

Set definition by intension can be written in the form $X = \{ x \mid P(x) = \text{TRUE} \}$, where P is a proposition specifying the properties that candidates for membership of X must satisfy to be members. Thus we have vowels = { x | x is a syllabic unit vowel OR x is an approximant }, which resolves the term 'vowel' at a lower level. Going down further levels we have vowels = { x | $x = a$ OR $x = e$ OR $x = i$ OR $x = o$ OR $x = u$ OR x is a lateral approximant OR x is a semi-vowel), and finally vowels = { x | $x = a$ OR $x = e$ OR $x = i$ OR $x = o$ OR $x = u$ OR $x = l$ OR $x = y$ OR $x = w$ }.

We say a proposition P is *well-defined* if there is an operational procedure for deciding of any potential candidate for membership, x, that $P(x)$ is *well-formed* in a logical sense, and that there is an operational procedure for deciding whether or not $P(x)$ is true. To illustrate this, consider the set X = { x | x is a house extension that is exempt from the building regulations }. The proposition $P(x)$ is true if "x is a house extension that is exempt from the building regulations". This requires an operational way of deciding if x is a house extension. Assuming that x is indeed a house extension, the operational procedures require the application of the following:

$P(x)$ = True according to the Building Regulations 1991 (as amended) if

x has a completely transparent or translucent roof
AND x has extension walls that are substantially glazed
AND x has a floor area not exceeding 30m squared.
AND x is sited at ground level.
AND x is permanently separated from the remainder of the property by means of a door.
AND x has separately controllable radiator (if fitted)
AND x has glazing satisfying the requirements of part N, Schedule 1 (toughened/safety glass).
AND x does not contain any drainage facilities. (i.e. sink, WC, or washing machine)
(Source: http://www.conservatoriesonline.com/planperm.htm, accessed 30-10-05)

Suppose x were an elephant. Then x fails the test that it is a house extension, and does not belong to the set of things exempt from building regulations. Suppose x is a 'my glass conservatory'. Then each of the conditions can be applied meaningfully to 'my glass conservatory' and decided to be true or false. In this case all of the sub-propositions needs to be true for $P(x)$ to be true.

Conventionally hierarchical structure is associated with classification and OR-aggregations. Related to this are the many methods of clustering concepts to form higher level constructs, *e.g.* houses and cathedrals are buildings. Star-hub analysis is very useful for this, but that is outside the scope of this paper. Important through the OR-aggregation is, this paper argues that AND-aggregations are absolutely fundamental in defining the vocabulary of multilevel systems.

6. PART – WHOLE PARADOXES AND RELATIONAL STRUCTURE

The relationship between parts and wholes causes endless unnecessary confusion unless it is made mathematically well-defined. For example, Figure 19 illustrates the conundrum that “Simpson’s Finger belongs to Simpson. Simpson belongs to the Philosophy Department. Simpson’s Finger belongs to the Philosophy Department”. This vernacular way of expressing things seems to produce the very odd conclusion that Simpson’s finger belongs to the Philosophy Department.

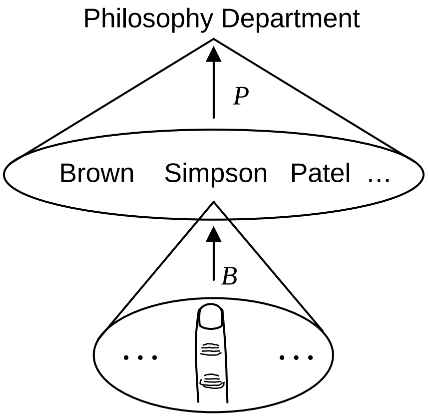


Figure 19. Simpson’s finger belongs to the Philosophy Department.

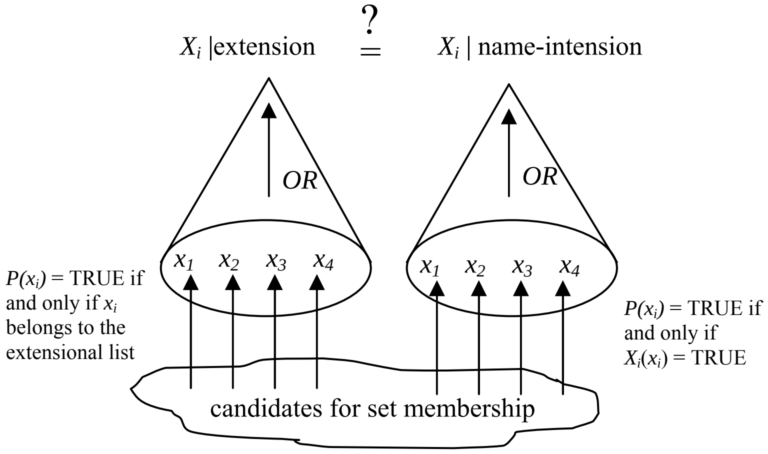


Figure 20 Set names define intensional proposition that may be inconsistent with an extensional definition of the set

Let the assembly relation R take a set of parts, S, to the higher level structure x, R: $S \longrightarrow R(S) = x$. Then define the *R-base of* x, $\text{base}_R(x)$, to be the set S, and write $\text{base}_R(x) = S$. We will also use the notation $R^{-1}(x) = S$, to suggest *disassembling* x into its constituent parts. In other words the *R-base* of a structure, x, is a set of component parts, S, at a lower level. The particular lower level and higher level are implicit in the relation R.

Suppose $R^{-1}(x) = S$, and $S = \{ s_1, s_2, s_3, s_4, \dots \}$. Suppose that there are component sets C_i and assembly relations R_i, with $R_i(C_i) = s_i$, for $i = 1, \dots, n$. Then we write $R_i^{-1} \circ R^{-1}(S) = R_i^{-1}(s_1, \dots, s_i, \dots, s_n) = R_i^{-1}(s_i) = C_i$. Using this notation, Simpson belongs to P^{-1}(Philosophy Department) and Simpson’s Finger belongs to B^{-1}(Simpson). Nothing surprising there. We also have Simpson’s Finger belongs to $B^{-1} \circ P^{-1}$ (Philosophy Department), which it indeed does. It multilevel systems there are many such compositions, and they reflect relationships, *e.g.* if Simpson hurts his finger he may not attend a departmental meeting, even though his finger normally has no impact on the department.

The ‘paradox’ of Simpson’s finger is artificial and arises from equating the relations B and P under the single term ‘belongs to’. The relation B only has meaning in the context of the other parts of Simpson’s body, because ‘Simpson’ is a meaningless construct unless all his parts are present. Similarly, the relation P is part of the *definition* of the philosophy department.

Generally a relation like P does two things. First it identifies the *set* of component parts, either by extension or by intension. Thus to form the simplex $\langle x_1, x_2, \dots, x_n ; R \rangle$ it is necessary to have *decision functions* $D_R : x_i \longrightarrow \{\text{True, False}\}$. When the sets are defined extensionally, the decision function will be of the form $D_R(x_i)$ = True if and only x_i belongs to a given list, X_i. When the sets are defined intensionally, the decision function will be of the form $D_i(x_i)$ = True if and only if $P_i(x_i)$ = True, where P_i is a proposition about x_i with an operational procedure for determining the truth value.

A common source of inconsistency and apparent paradox can arises when sets are given names that attempt to describe their members (Figure 20). For example, a television programme classification scheme[6] had the class ‘sports not requiring equipment’, and gave boxing and wrestling as examples. Unfortunately this produces an inconsistency since $\{$ x | x is a sport not requiring equipment $\} \neq \{$ boxing, wrestling, …. $\}$ because both boxing and wrestling do require equipment.

When a set has an operational definition for determining its members in terms of lower level sets its defining proposition will be said to be *instantiated.* When there is an operational definition determining some of the elements the set will be said to be *partly instantiated*. Once the component sets are instantiated, P has to test if the n-ary relation holds between the vertices. Generally this is much more difficult than testing for parts. If the relation holds, P is said to be instantiated.

7. BACKCLOTH AND TRAFFIC

Apart from relational structure, most multilevel systems have patterns of numerical properties distributed across them, as illustrated in Figure 21. Generally the numbers at lower level aggregate up the hierarchy. This can be by simple linear addition, as shown between *Levels N* and *N+1*, or by non-linear functions, as illustrated between *Levels N+1* and *N+2*. Generally the relational structure of systems is relatively fixed, while the numerical values are relatively dynamic. For this reason we refer to the relational structure as the *relatively static backcloth* and say that it supports the *relatively dynamic traffic* of activity on the system. For example, the topology of a motor car is relatively fixed, while its speed may change considerably over time. Similarly, the infrastructure of the stock market involves many relatively fixed relationship, and this acts as a backcloth supporting the highly dynamic traffic of trades and prices.

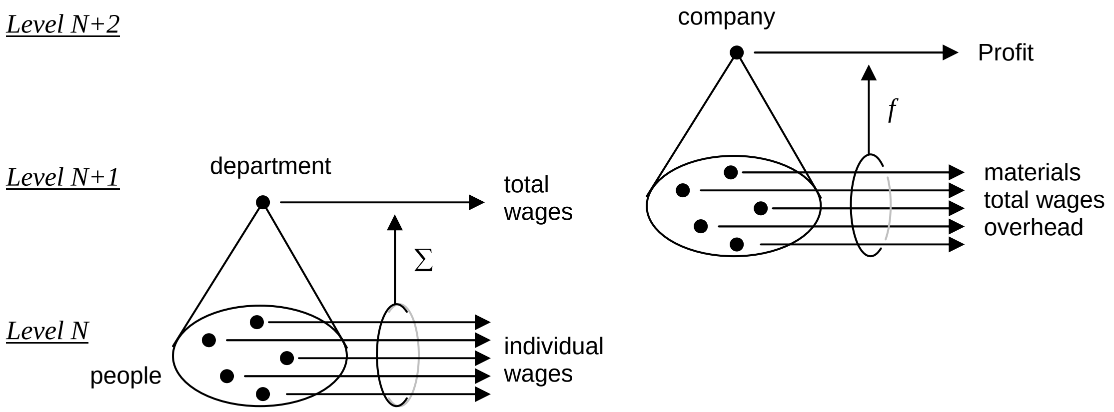


Figure 21. Multidimensional traffic on the multidimensional backcloth[4].

Suppose that a system has been designed and that there is a blueprint with sufficient information to fabricate it. Then that blueprint will specify all the necessary parts at *Level N* and explicitly give all the assembly relations enabling the atomic parts to be built into intermediate level structure or components at *Level N+1*. Similarly the blueprint will give all the assembly information necessary to assemble the system through all the levels, in a bottom-up fashion.

This brings us to the fundamental question: where do the 'right' atomic sets come from, and where do the 'right' assembly relations come from? These are scientific questions, of the same kind that Galileo answered by defining sets of time intervals and distances along an inclined plane, and that Harvey answered by defining sets of arteries and veins. In neither case was defining the 'right' sets a trivial matter, both having occupied other scientists for centuries. In both cases the breakthrough came from postulating the nature of the relations that assemble the parts into the whole. Thus in formulating a theory about a multilevel system, the scientist has to find an answer to the *Intermediate Word Problem*, illustrated in Figure 22.

The Intermediate Word Problem reflects the *top-down* aspects of understanding systems. If the system already exists, one looks at it and tries to see how it is made up of parts.

This *reductionist* approach, contrary to complaints that wholes cannot be understood in terms of their parts, is an essential part of the scientific process. If it were otherwise it would mean that no information about parts of the system would be useful in understanding it. If this were so the system would only have one relevant level, that of the 'The System'. Some systems are like this, *e.g.* the volume-temperature-pressure relationships on a 'fixed mass of gas system'. This system could be represented by a cone, with the set of gas molecules in an Euler circle aggregating into 'fixed mass of gas system', and in principle knowledge of the molecules could be used to calculate the volume, temperature and pressure. In practice the lower level structure is ignored, because the Gas Laws do not require them to be explicit.

8. TOP-DOWN AND BOTTOM-UP REASONING IN DESIGN

If the system does not already exist, the designer has to suggest the parts and how they aggregate between levels. In this paper we will describe this process for design, and argue that it is also the process that has to be used in formulating a science of artificial systems.

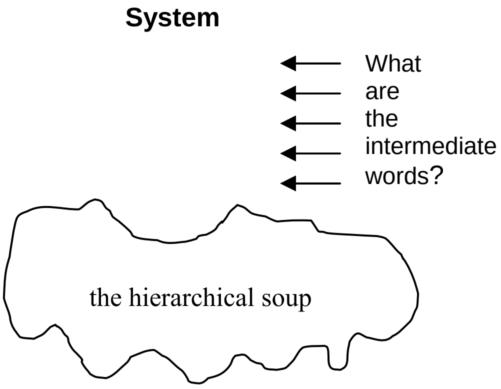


Figure 22. The Intermediate Word problem

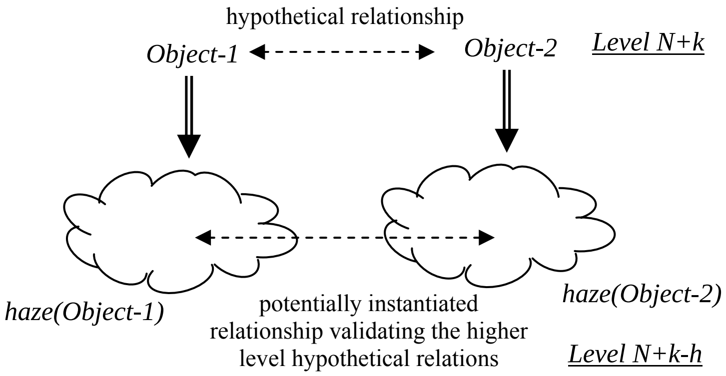


Figure 23. A top-down *hierarchical haze*

The *bottom-up* part-whole assembly approach is complemented by a top-down approach in which designers *analyse* systems in terms of their parts and characteristics. To do this the designer addresses the Intermediate Word Problem (Figure 22). When we look at any system as a whole, our impression of it is a Gestalt. Alongside this, there may be prior knowledge of the system, and we are usually able to see the constituent parts and substructures.

Thus we experience the whole in some sensory way, and we make associations between the whole and pre-existing things in our minds. The prior knowledge that we have forms a ‘soup’ of information, “a pre-logical primordial source containing the building blocks of all subsequent substructures” [6]. Thus, when analysing a system, the designer has to abstract a vocabulary to represent the system between the uninformative highest level term, “the system”, and this vernacular hierarchical soup (Figure 22).

As a design progresses, some things will be totally instantiated at lower level, but some will not. Those things that are not instantiated are *hazy* abstractions, important parts of the design awaiting further information. Even though a part of the system is not instantiated, it may play an important part in reasoning about the system. Relationships are hypothesized between objects whether or not they are instantiated. These hypotheses may or may not be validated as the design becomes grounded and higher level constructs are instantiated in terms of testable lower levels (Figure 23).

These idea are illustrated in Figure 24 for a helicopter. Here the engineer has identified parts of the system and given them *names*. These names can then become part of a formal vocabulary used to describe the system and reason about it. In this hierarchical decomposition, the lowest level is *grounded* in the soup.

Any designed and manufactured object can be represented in terms of a hierarchy of assembly like this. The use of numbers to define levels requires some justification, because this type of hierarchy may not be linear but more like a tree or a lattice. For our purposes it is sufficient to note that if *x* aggregates into *y*, then *x* is at a lower level than *y* in the representation. Thus when the designer begins, even though nothing may have been decided in terms of actual component parts or assemblies, he or she knows that when they have finished the system will be represented in terms of a specification of its parts and precise instructions on how to assemble them at every level. This is the *blueprint* that the designer hands over for fabrication. The blueprint communicates the necessary and sufficient information to build the system. When it is complete every part in the hierarchy is explicitly identified and named.

The design process can be viewed as the process of creating the blueprint. Initially the designer starts with some hazy idea of the ‘the system’ at some high level, in the context of a soup of terms and prior knowledge from which the explicit vocabulary of the design must be abstracted (Figure 22). On reflection the designer will postulate various subsystems and components as intermediate words. Some of these may be completely unspecified in terms of their details, and they are represented in Figure 23 by hazy cloud-like symbols. Some parts of the system may be entirely specified from the start, such as the tail rotor in Figure 25(a). For example, this may be a legacy subsystem that must be used in the design as part of the specification.

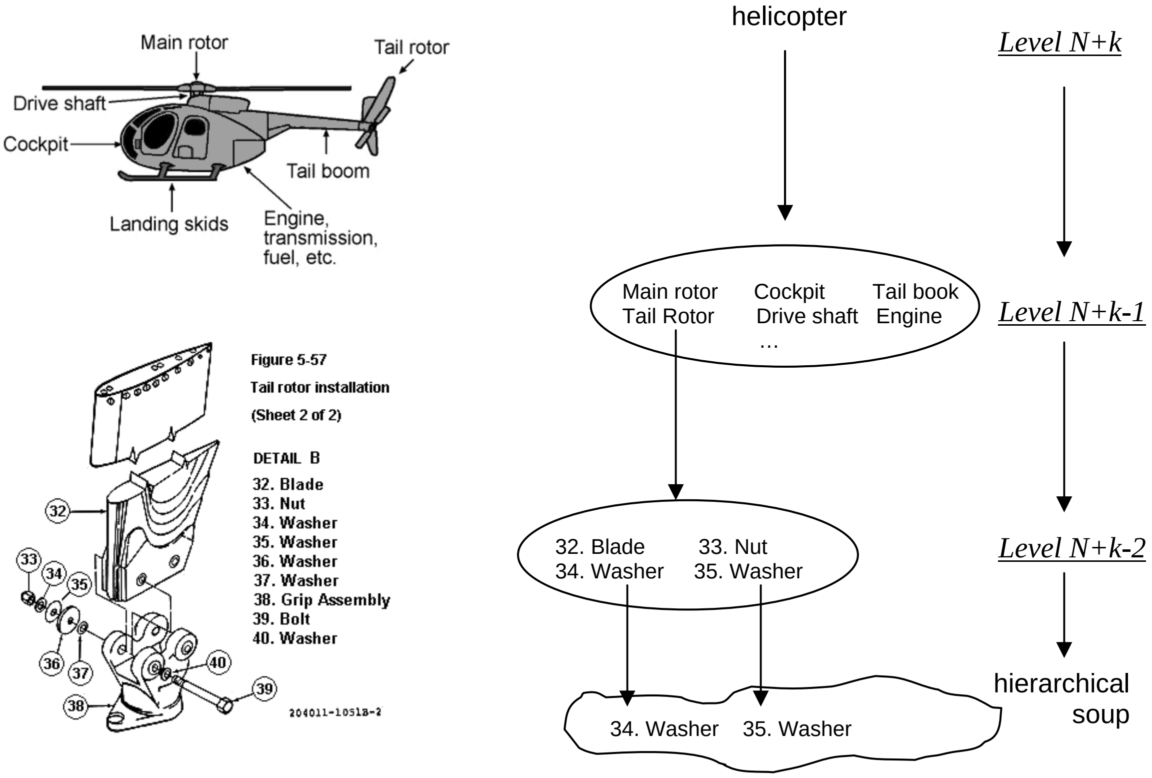
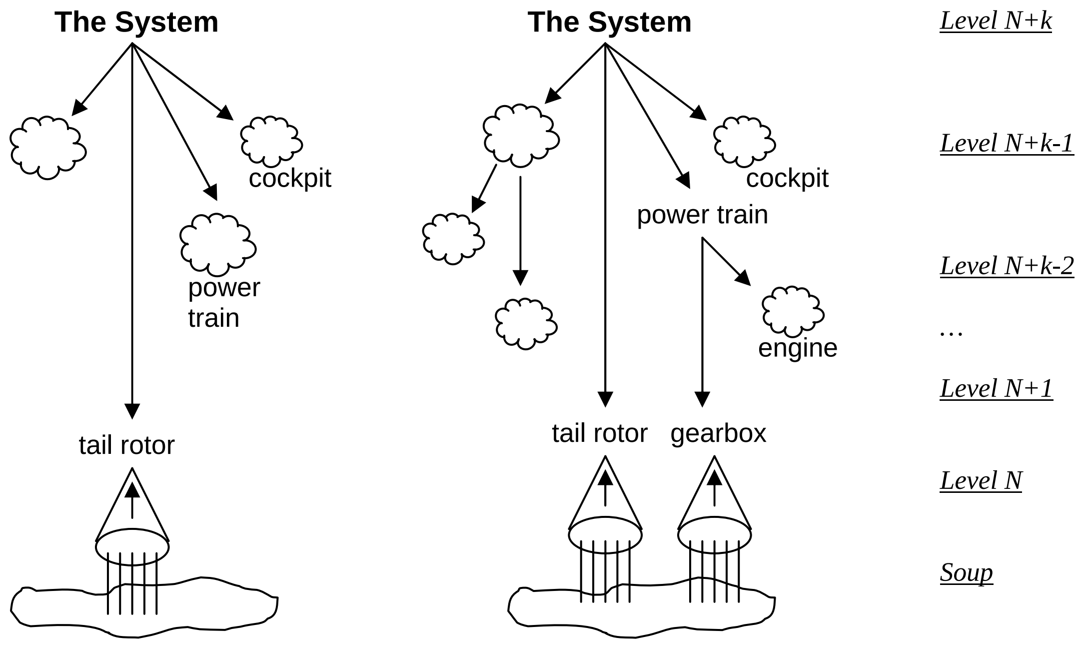


Figure 24. Abstracting intermediate words for a helicopter (Source: Johnson, 2005)



(a) Initial sketch (b) partial instantiation

Figure 25. Design as the process of building an ontology (Source: Johnson, 2005)

As the design proceeds, the hazy subsystems may be worked up in more detail. This could include postulates of other hazy clouds at lower levels, or it could include fully instantiated components, as illustrated by the gearbox in Figure 25(b). In principle the design proceeds by the designer working down the hierarchy, adding more explicit detail until the whole design is instantiated by particular components, and subsystems built from explicitly specified components and sub-assemblies.

Some part of the design process may be bottom-up, as the designer creates new assemblies with the necessary properties to be components at higher level. In practice one can imagine designers constantly scanning the design in both top-down and bottom-up modes, trying to connect the higher level abstractions to the lower level realities.

As the design proceeds and more of the hazy clouds become instantiated with fully defined objects, implicit assumptions may turn out to be problematic. For example, in Figure 25(b), the particular the rotor and the particular gearbox may have undesirable interactions. This unexpected emergent property becomes a design problem. Perhaps the designer can find some new way of putting the components together that obviates the problem. If not, it may be necessary to back-track, abandoning the particular tail rotor or the particular gearbox. In extreme cases the original specification that the particular tail rotor must be used may have to be changed, so that the fully specified component has to be replaced by a hazy cloud at a higher level in the representation.

9. THE DESIGN CYCLE

The design process has been described as that of building a multilevel language to represent the artefact being designed, with top-down – bottom-up interactions until both meet and a consistent blue print is achieved. In practice the process is more complicated than this. The design process begins with a perceived need or *requirements*. These are translated into a *specification* which the designed system should meet. Even at the blueprint stage, when fully instantiated, the design may be evaluated and found deficient. Thus everything said so far has to be put in the context of the *design cycle*. There are many models of the design cycle in the literature, but they all have the generate-evaluate loop shown in Figure 26.

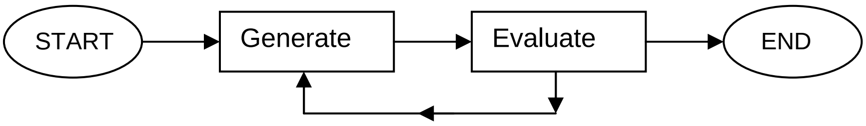


Figure 26. A simple representation of the design cycle

In practice the design process is even more complicated than this. Design can be thought of as a search for a solution to the problem “find a systems that satisfies these specification”. Very often the specifications are over-constrained or under constrained, so there are either no solutions, or too many solutions. In the former case the specifications have to be relaxed, with some constraints being removed or eased. For example, some specified property may be abandoned as desirable but not essential, or cost constraints may be eased by agreeing a higher price. In the latter case, new constraints may be added to force a solution, usually seeking a more optimal solution. Thus the goalposts are moved, and any future solution will be a solution to a different problem to that originally specified. In this one can see that the design process is a co-evolution between specification and proposed solution, until a problem-solution pair is found that is considered to be satisfactory.

In the context of the previous sections, it would be rare for a complete design to be evaluated in fully instantiated form and rejected. This is because the design process can be very expensive in terms of people’s time and the collection of information at various levels throughout the process. Often designs or parts of designs can be rejected as the design process proceeds, with the focus at any time being on the most critical parts, however defined.

10. CONCLUSIONS

The majority of this paper has been devoted to sketching out a mathematical theory of multilevel systems and its related multidimensional structure. The whole enterprise is based on what was called the *Fundamental Construction of Multilevel Systems*, where a set of component parts at a given hierarchical level is assembled into a whole at a higher hierarchical level, in bottom-up fashion. In principle this process is grounded at the level of concrete objects and builds up increasingly more complicated structure in a combinatorial fashion. In contrast to this 'put one brick on top of another' approach is a top-down approach in which the system is dissembled into its parts. When the system does not exist, as it often does not in design, the top-down approach is dealing with abstractions. These are instantiated at lower level in a hazy way, and all reasoning at the higher levels is contingent on lower level instantiation.

A theory of design has been proposed in which the designer creates new artefacts and the multilevel vocabulary to describe them. Initially the design is ill-specified, being sketched in terms of hazy high-level constructs and some more concrete structures at lower levels. The design proceeds by hypothesis-making at all level, and periodically subjecting hypotheses to validation as the design is instantiated. The generate-evaluate cycle operates at all levels, with a premium on rejecting incorrect hypotheses before too much resource is expended on consequent structures. The outcome of this design process is a fully instantiated blueprint, in which the vocabulary necessary to describe the system has been built, and all hypotheses about the system have been tested, leading to a 'theory' of the system.

In this context, it has been argued that the designers of new systems are the first scientists to accumulate knowledge about those systems. Furthermore, it has been argued that scientists exploring artificial systems must follow the path trodden by designers, especially if they have ambitions to change and control those systems as engineers, planners, or administrators. In summary, it has been argued that, if design is the science of the artificial, then the science of the artificial is design.

ACKNOWLEDGMENTS

The ideas in this paper have progressed considerably as a result of being able to discuss them with members of the Embracing Complexity in Design Cluster. I am grateful to other members of the cluster for many fascinating discussions on complexity and multilevel systems in design, and to both the Arts and Humanities Research Council and Engineering and Physical Science Research Council for funding the cluster.

REFERENCES

1. Simon, H. *The sciences of the artificial*, MIT Press (Cambridge, Mass), 1969.
2. Johnson, J.H., 'Relational structure in planning and design', *Environment and Planning B: Planning and Design* Anniversary Edition, 57-67, 1998.
3. Barabási, A-L, *Linked*, PLUME (Penguin Group, New York), 2003.
4. Johnson, J.H., 'Exploring Complexity in Collaborative Design Solutions', Proc. Int. Workshop on Collaborative Design, Qin and Johnson (eds), Brunel University, 23-24 June 2005.
5. Johnson, J.H., 'Combinatorial Structure in Digital Pictures', NERC Remote Sensing Project Discussion Paper 1,Mimeo, Centre for Configurational Studies, The Open University, January 1985.
6. Gould, P., Johnson, J.H., Chapman, G.P., *The structure of television*. PION (London), 1984.
7. Johnson, J.H., 'Some structures and notation of Q-analysis', *Environment and Planning B*, **8**, 73-86, 1981.
8. Johnson, J.H., 'A language of structure in the science of complexity', *Complexity*, **1**(3), 22-29, 1995.
9. Johnson, J.H., 'Q-transmission in simplicial complexes', *I.J. Man-Machine Studies*, **16**, 351-377.

Designing in the real world is complex anyway - so what? Systemic and evolutionary process models in design

Wolfgang Jonas
University of Kassel, Menzelstr. 13-15, D-34121 Kassel, Germany
jonasw@snafu.de

ABSTRACT

Designing is a heterogeneous, fuzzily defined, floating field of various activities and chunks of ideas and knowledge. Available theories about the foundations of designing as presented in "the basic PARADOX" (Jonas and Meyer-Veden 2004) have evoked the impression of Babylonian confusion. We located the reasons for this "mess" in the "non-fit", which is the problematic relation of theories and subject field. There seems to be a comparable interface problem in theory-building as in designing itself.

"Complexity" sounds promising, but turns out to be a problematic and not really helpful concept. I will argue for a more precise application of *systemic* and *evolutionary* concepts instead, which - in my view - are able to model the underlying generative structures and processes that produce the visible phenomenon of complexity. It does not make sense to introduce a new fashionable meta-concept and to hope for a panacea before having clarified the more basic and still equally problematic older meta-concepts.

This paper will take one step away from "theories of what" towards practice and doing and try to have a closer look at existing process models or "theories of how" to design instead. Doing this from a *systemic perspective* leads to an *evolutionary view* of the process, which finally allows to specify more clearly the "knowledge gaps" inherent in the design process. This aspect has to be taken into account as constitutive of any attempt at theory-building in design, which can be characterized as a "practice of not-knowing".

I conclude, that comprehensive "unified" theories, or methods, or process models run aground on the identified knowledge gaps, which allow neither reliable models of the present, nor reliable projections into the future. Consolation may be found in performing a shift from the effort of *adaptation* towards strategies of *exaptation*, which means the development of stocks of alternatives for coping with unpredictable situations in the future.

Keywords: Design process models, complexity, systems, evolution

1. FOREWORD: COMPLEXITY – SOME SCEPTICAL REMARKS

"Complexity" has been one of the buzzwords in design and design theory for at least 10 years now. Design is facing "the challenge of complexity", design is "embracing complexity", and so forth. Complexity theory is promoted as the new meta-tool for dealing with complexity. But what is complexity? Is complexity in design the same as complexity in complexity theory? This would make things much easier. One may solve this question, as for example Bar-Yam (1997) does at the very beginning of his seminal book by defining:

- *complex = consisting of interconnected or interwoven parts / not easy to understand or analyze, and*

- *complexity = the amount of information needed to describe it.*

These are perfect definitions with regard to formalized approaches and algorithms, as in cellular automata or in well-defined multi-agent systems. But - to give a simple example - what is the amount of information needed to describe the emotional relation of a user and his/her object of desire, which may be essential for the success of a new product.

John Horgan, in his June 1995 *Scientific American* editorial entitled "From complexity to perplexity", mentions 31 definitions of complexity and states the lack of a "unified theory". Mikulecky (2003) follows Horgan and argues that complexity is the result of the failure of the Newtonian Paradigm (which represents the world as simple mechanisms) to be generic:

"Complex systems and simple systems are disjoint categories that encompass all of nature. The world therefore divides naturally into those things that are simple and those things that are complex. The real world is made up of complex things. Therefore the world of simple mechanisms is a fictitious world created by science or, more specifically, by physics as the hard version of science. This is the world of the reductionist. It is modelled by the Newtonian Paradigm and simply needs sufficient experimentation to make it known to us. Those experiments involve reducing the system to its parts and then studying those parts in a context formulated according to dynamics. …"

The way science is done is the modelling relation. We observe the world around us and try to make sense out of that sensory information by calling the events that make it change as we observe causality. We encode the real-world system into another system, a formal one, which is completely under our control. Once we think we have an appropriate formal system and have found an implication that corresponds to the causal event in the real world, we must decode from the formal system in order to check its success or failure in representing the causal event.

This worked for a long time and is tremendously successful. But observers came up with aspects that the Newtonian Paradigm failed to capture and a new explanation was required. Mikulecky (2003):

"Complexity was born! This easily can be formalized. It has very profound meaning. Complexity is the property of a real world system that is manifest in the inability of any one formalism being adequate to capture all its properties. It requires that we find distinctly different ways of interacting with systems. Distinctly different in the sense that when we make successful models , the formal systems needed to describe each distinct aspect are NOT derivable from each other."

Irreducible "knowledge gaps" are showing up, and there will probably be no "unified theory" of complexity. This is why I recommend to skip the concept of complexity (or to use it as a metaphor denoting our inability) and turn back to the older concepts of *system* and *evolution*.

2. DESIGN PROCESS MODELS – A GENEALOGICAL SKETCH

Stated in the most general manner, a design task consists in transferring an existing state of a "system" into a preferred one, whereby "system" will normally be considered as some kind of complex (!) "whole", consisting of elements and relations between these elements. The preferred state can be defined as an optimal "fit" of the system or artefact and its environment. The artefact is what designers design, whereas the environment consists of the constraints that have to be met and which cannot be directly controlled by design. The "interface" region between the artefact and the environment is the "location" of design activities (Alexander 1964, Simon 1969).

The system concepts in design as used in "complex systems" appears to be rather simplistic. There is hardly any reference to the elaborate thermodynamic and biological theories of open / dissipative / closed systems, which explain how systems are able to keep a state of high order far from equilibrium, thus temporarily overcoming the 2nd law of thermodynamics. Systems concepts in design are mainly based on simplified applications of Wiener's *cybernetics* (Wiener 1948), dealing with mechanisms of *feedback, communication and control* in goal-oriented processes. He explicitly warned of any hope that his approach could contribute to the healing of the diseases of society.

Weaver's concept of *"organized complexity"* (Weaver 1948) filled the obvious gap between the classical Newtonian concepts of "problems of simplicity" and "problems of disorganized complexity" and might have provided a powerful basis for dealing with complex social problems / design problems. This was his enthusiastic programmatic appeal for the next 50 years; but his approach was neglected in favour of *computability*.

Operations Research (OR) can be regarded as the first application of systems thinking in "designerly" processes, such as planning and engineering, from the 1950s onward. The problem-solving process in OR consists of the definition of the solution space, the formulation of the measure of merit, the fixing of constraints, and the optimisation process, leading to a local or global optimum. This was tremendously successful, as e.g. the big NASA projects prove.

The design methods movement in the 1960s adopted and developed these approaches. Symbolic models of the design problem have to be built, consisting of factors, which describe the problem situation and causal relations between these factors. Ideally, the solution criterion is given in a quantitative manner, as a measure of merit function (even aesthetic criteria have been treated in this way, as we know). Numerical optimisation methods based on closed mathematical calculus or iterative heuristic algorithms can be applied to this problem type. The problem space has to be limited and well-known, and if the problem is properly stated, then the solution is just a re-formulation of the problem, or a change of representation, which can be carried out by means of the same symbolic language that was used for the problem definition (Simon 1969). But unfortunately, this has never been the normal situation in design.

Design problems may be categorized according to the parameters *problem / solution space* (elements + relations available), which can be either limited or open, and *solution criterion* (measure of merit), which can be either quantitative or qualitative, which means influenced by ethical and aesthetic factors. Entirely numerical solutions are possible, if the solution space is *limited* and the solution criterion is of *quantitative* nature, which is the case, for example, in a chess problem or in the optimisation of a streamlined shape according to aerodynamic criteria. In all other cases we have value-laden solutions of ethical or aesthetic nature (even the apparently highly quantitative problem to bring a man to the moon has a large number of qualitative subtasks, as for example the interior human-machine interface of a vehicle). Value-based decisions of minor or major impact have to be taken at various moments during the solution process.

As soon as the relevant environment of a design problem is no longer natural, but influenced by psychic or social aspects, then the concept of *time* in the process is changing. Systems have memories and imagination. Time is no longer a linear parameter, the "fourth dimension", but the source of uncertainty. The future can be conceived as a *projective space*, determined not only by natural trajectories, but by plans, wishes, hopes, fears, decisions, etc. In other words: it is a space of imagination. The development of psychic and social situations is proceeding in highly unpredictable ways; the fit between the artefacts and the environments will probably disappear before long.

What about remaining prediction capabilities for the *future* fit of solutions in *non-natural* contexts? The question comprises the issue of *"how do we want to live?"*, and marks the shift from "first-generation" to "second generation" methodology, which is closely connected to Horst Rittel (1972). In his view, first-generation methods seem to start once all the truly difficult questions have been dealt with already. He introduced the notion of "wicked problems" and tried to denote the *limits of rationality* related to this kind of problems. Rational behaviour means the attempt to foresee the consequences of intended actions, which results in various dilemmas and paradoxes, for example that the more rational one is in discovering the causal chains of future consequences of interventions, the more one is disabled to act.

According to Rittel, these dilemmas have to be overcome by opening up the closed algorithmic problem solving process and initiating a process of *argumentation* and *negotiation* among the stakeholders instead. In other words: he suggests a change from 1st order cybernetics to 2nd order cybernetics: not systems are observed, but systems observing systems. This introduces, as a central new part, the *design of the "problem" itself*. Under conditions of second order observation we have to account for the fact, that the problem itself is not "given", but has to be constructed by the stakeholders. In consequence, problems are changing their character in the course of the solution process. No information is available, if there is no idea of a solution, because the questions arising depend on the kind of solution, which one has in mind. One cannot fully understand and formulate the problem, before it is solved. Thus, in the end, the solution is the problem. Therefore Rittel argues for the further development and refinement of the argumentative model of the design process and the study of the logic of the designers' reasoning, where logic means the rules of asking questions, generating information, and arriving at judgements.

In view of this situation Rittel (Cross 1984: 326) states in his slightly ironic manner:

> "All of which implies a certain modesty; while of course on the other side there is a characteristic of the second generation which is not so modest, that of lack of respect for existing situations and an assumption that nothing has to continue to be the way that it is. That might be expressed in the principle of systematic doubt or something like it. The second-generation designer also is a moderate optimist, in that he refuses to believe that planning is impossible, although his knowledge of the dilemmas of rationality and the dilemmas of planning for others should tell him otherwise, perhaps. But he refuses to believe that planning is impossible, otherwise he would go home. He must also be an activist."

John Chris Jones (1970) puts it more general and metaphoric, when emphasizing the necessity of designing the design process itself. A considerable part of the design capacities has to be re-directed from the problem to the process. The designer as "black box" (the artist) as well as the designer as "glass box" (the follower of 1st generation methods) have to change their attitude towards a self-conception of *designer as "self-organizing system"*, who is observing the evolving artefact plus himself observing the evolving artefact.

3. INHERENT PATTERNS – CIRCULARITY AND AUTOPOIESIS

Circularity as a characteristic of problem-solving and purposive design processes is showing up. We know DO - loops as instructions for iterative processes in formal languages in software-programming. We know the TOTE - scheme (Test - Operate - Test - Exit) from cognitive psychology (Miller et. al. 1960) as the prototypical pattern for dealing with iterative heuristics and feedback in design methods. Most of these design methods consist of linear sequences of steps of specific subtasks plus TOTE cycles for the necessary feedback. Opaque systems, called *"black-boxes"* are rendered "white" and manageable by means of circular feedback-models. Human agents act as detached operators of these "machines". Thus systems have been typically treated mechanistically as open (for matter, energy and information), and in interaction with their context, transforming inputs into outputs as a means of creating the conditions necessary for survival. Changes in the environment are seen as input stimuli, to which the system must respond in defined manners.

The concept of *autopoietic closure* in living and meaning-based systems is essential for the further argument concerning design processes. Autopoiesis characterizes the self-referential logic of self-(re)producing systems. Maturana and Varela (1985) argue, that living systems are organizationally closed, i.e. without any input or output of control information. Operations only refer to themselves and the system's internal states. The impression, that living systems are open to an environment, results from an attempt to make sense of such systems from the perspective of an outside observer. If at all, "black boxes" can only temporarily be "whitened" by means of an interaction of observer and observed (Glanville 1982). The aim of autopoietic systems is ultimately to maintain their own identity and organization. A system cannot enter into interactions that are not specified in the pattern of relations that define its organization. In this sense the system's environment is really a part of itself. The theory of autopoiesis thus admits that systems can be recognized as having "environments", but insists that relations with any environment are *internally* determined; systems can evolve only along with *self-generated* paths.

The theory of autopoiesis encourages us to understand the transformation of living systems as the result of internally generated change. Rather than suggesting that the system merely *adapts* to an environment or that the environment *selects* the system configuration that survives, autopoiesis places principal emphasis on the way the total system of interactions shapes its future and evolves. Autopoiesis presents a modification of Darwinian theory: while recognizing the importance of system variation and the retention of "selected" features in the process of evolution, the theory offers different explanations as to how this occurs. Changes are eventually induced, but not directed by means of perturbations from outside. The emphasis is shifting from adaptation of a system to its environment towards *co-evolution of autonomous systems*.

Morgan (1986: 245) was one of the first to apply the biological concept of autopoiesis to a design-related field, namely organization theory:

" When we recognize that the environment is not an independent domain, and that we don’t necessarily have to compete or struggle against the environment, a completely new relationship becomes possible. For example, an organization can explore possible identities and the conditions under which they can be realized. Organizations committed to this kind of self-discovery are able to develop a kind of systemic wisdom. They become more aware of their role and significance within the whole, and of their ability to facilitate patterns of change and development that will allow their identity to evolve along with that of the wider system."

This is a very positive interpretation of autopoiesis, and probably a step forward with respect to the problems of organizations. But it still neglects the fact that the environments of autopoietic systems consist of various other, equally stubborn autopoietic systems. Luhmann (1984) has formulated this radical generalization of biological autopoiesis. He extends it for the purpose of describing mental and social systems. His theory of social systems provides more delicate instruments for an identification of the problem and a composed deconstruction of unfounded expectations in design theory. Organizations, as described by Morgan, are one of several sub-categories of communicative / social systems, all of which are operationally closed, autopoietic systems. Living systems act in the medium of life, mental systems in consciousness, and social systems in communication. Both mental and social systems operate with *language* and *meaning*. Communication cannot happen without presupposing consciousness and vice versa, nevertheless both are closed, without any transfer of information. Language, which Luhmann calls a "variation mechanism of socio-cultural evolution", is the ultimate instrument for coupling mental and social systems. This strange, fuzzy, non-causal coupling, called *interpenetration*, seems to be the most powerful driver of human evolution and learning.

4. EVOLUTIONARY EPISTEMOLOGY – RECOGNITION AND EXPLANATION

The epistemic characteristic of design can be assumed as *learning process*. This process can be considered as biological, grounded in the need of organisms to survive in an environment. The aim cannot be final "true" representation of some external reality, but rather a process of *(re-) construction* for the purpose of appropriate *(re-) action*. Yet Aristotle suspected, that the recognizability of the world must rely on the fact, that there is a kind of similarity between the "particles" of the world and those in our senses. The history of biological evolution suggests similarities of the way the material world is structured and the way we think of it. Evolutionary epistemologists (Campbell 1974) argue that the Kantian transcendental *a priori* has to be replaced by the assumption of an *evolutionary fit* between the objects and the subject of recognition.

The evolutionary model of knowledge production presents a scheme with structural identity from the molecular up to the cognitive and cultural level (Riedl 2000). The basic structure reveals a circle of trial (based upon expectation) and experience (leading to success or failure, confirmation or refutation), or of action and reflection. Starting with passed cases, the circle consists of an inductive / heuristic semi-circle with purposeful learning from experience, leading to hypotheses and theories and prognoses about how the world works, and a deductive / logical semi-circle, leading to actions and interventions, which result in the confirmation or refutation of theories due to new experiences, etc. Internal or external perturbations (called ideas, creativity, curiosity, … or accidents, environmental changes, …) influence the circle, leading to stabilizations (negative feedback) or amplifications and evolutionary developments (positive feedback).

Only very recently in the cultural evolution this general scheme was split into the "ratiomorphous" (the term was coined by Konrad Lorenz) systems of recognition and the rational systems of explanation / understanding, with its most extreme form: the logical positivist dualism of "context of discovery" (acting) vs. "context of justification" (thinking). While the ratiomorphous process of recognition has a high potential in dealing with complex, evolving phenomena, it is not always useful for causal explanations, and vice versa. But this "dilemma" is not inherent in the nature of knowledge production, but rather a consequence of the dualistic concept, which we have imposed on the process. The path from recognition to explanation is continuous and circular, sometimes with dead ends. Our language is too poor, or, too much locked in the "black&white" tradition, to express the beautiful transitory shades of "grey" between the poles.

Table 1. Recognition vs. Explanation (Riedl 2000: 53 – 55).

Recognition (Erkennen)	**Explanation (Erklären / Verstehen)**
- networks, many causes	- linear cause – effect relations
- simultaneous (simul hoc)	- sequential (propter hoc)
- 4 Aristotelian causes considered	- only causa efficiens considered
- only local validity, context is crucial	- global validity claimed, context excluded
- allows no experiments, mostly irreversible	- relies on experiments, mostly reversible
- prognosis is projection	- prognosis is forecasting
- correspondence of organism / artefact in a milieu	- coherence of elements inside a system
- reaches into high complexity	- reduces complexity
- fitness, “truth” means strong design	- “truth” means correct causal relations
- is labelled “pre-scientific”	- is labelled “scientific”

The argument of *naturalized epistemology* appears in various forms. John Dewey (1986) argues that processes of circular action, driven by intention, are the essential core of knowledge generation. The separation of thinking as pure contemplation and acting as bodily intervention into the world becomes obsolete; quite the reverse: Thinking depends on real world situations that have to be met. Thinking activity is initiated by the necessity to choose appropriate means with regard to expected consequences. The projected active improvement of an unsatisfactory, problematic situation is the primary motivation for thinking, designing, and, finally - in a more refined, purified, quantitative manner - for scientific research and knowledge production. According to Dewey, knowing is a manner of acting and "truth" is better called "warranted assertibility".

To come back to design: Schön´s (1983) epistemology of "reflective practice" can be regarded as the design-related description of these concepts. It is this unspecific pattern, which Cross (2001) characterizes as "designerly ways of knowing":

"The underlying axiom of this discipline is that there are forms of knowledge special to the awareness and ability of a designer, independent of the different professional domains of design practice."

Evolutionary epistemology uses the most basic generative mechanism to explain learning in the living world, thus explaining the ongoing production and re-production of both artefacts and knowledge, finally of design and science as dynamic forms. This is the "essence", and there is no need for any specific nature of knowing in design. The theory of socio-cultural evolution seems to be a useful framework to denote the unpredictability of design developments and project outcomes, thus the limits of causal explanations, in a scientific manner.

5. SOCIOCULTURAL EVOLUTION – APPLICATION TO DESIGN

Autopoietic systems show a high independence from internal and external perturbations (negative feedback compensates for the irritations). On the other hand it is one of the insights of chaos theory, that circularity in simple mathematical models, can cause so-called *deterministic chaos.* Minimal differences in initial conditions of the system parameters can cause completely different outcomes, so that predictability of final states is lost (positive feedback amplifies perturbations and triggers evolutionary change). Natural evolutionary patterns of development, with their sequence of stable phases and sudden variations seem to be based on an interplay of negative and positive feedback mechanisms. The evolution of artefacts shows similar patterns (Fig. 1).

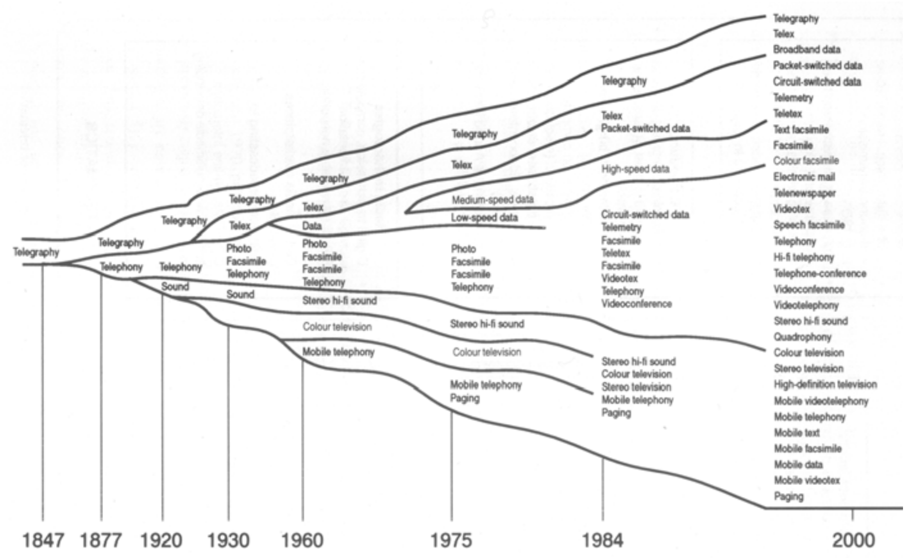


Figure 1.4 The expanding range of telecommunications services: prospects for the year 2000

Source: Adapted from Ungerer, 1988

Figure 1. Evolution of artefacts (Graham and Marvin 1996).

Hybs and Gero (1992) describe artefacts as entities struggling for the survival of the fittest in the hostile environment of the market; but the approach is still sub-complex. We (seem to) know where we come from, but we do not know, where we are going. At least we know the ancestors of our current artefacts, which means some interpretation capacity for design history. Nevertheless we normally do not know the influences that acted upon the bifurcation situations and resulted in exactly this and no other development.

Also representations of design processes reveal these patterns (fig. 2). The nicely cut branches after the bifurcation points suggest, that there is a rational means to overcome the indeterminacy, to take a decision, which provides more than a random chance, that the decision is viable in the future. Rittel (1971/72: 48, 54, translation W.J.) comments this laconic:

"Constrictions are not 'natural conditions' but deliberate restrictions of the variety of solutions, mostly implicit signs of resignation. …

... In reality there is no opposition / sharp conflict between an … intuitive approach to solve a problem and … a controlled, reasonable and rational approach. The more control one wants to exert, the more well-founded one wants to judge, the more intuitive one has to be.

The endpoints in the more and more ramifying tree of causal explanations are always spontaneous judgements."

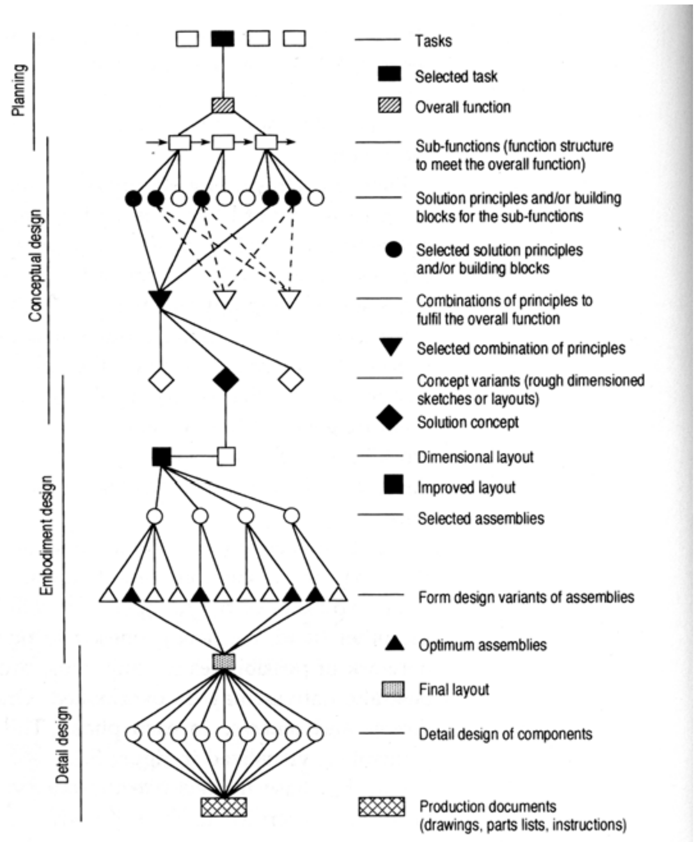


Figure 5.12 Divergence and convergence in the design process. The shadowed elements indicate the chosen points of departure for the next phase (source: VDI 2222).

Figure 2. Bifurcation cascades in the design process (Roozenburg and Eekels 1991).

These evident analogies in the processual patterns of natural and artefact evolution suggest the application of basic evolutionary concepts to the design of artefacts. No 1:1 analogies are sought; of course variation in a meaning-based context is different from variation in organisms. Thus, if we are aiming at new descriptions and tools for the design process, we have to identify the elements and processes of natural evolution, which can be transferred to the evolution of artefacts. We should focus on the problem of increasing the probability of success with respect to a decision to be taken.

Luhmann's theories are closely related to evolutionary epistemology. In his main oeuvre (1997) he has started to work out the concept of *social evolution*. Evolution theory is based upon the *system / environment* distinction; it is this difference, which enables evolution. Evolution theory does not distinguish historical epochs, but the circular sequence of *variation*, *selection*, and *re-stabilization*. It serves for the unfolding of the paradox of "the probability of the improbable". Re-stabilization is essential, because it is the condition for variation and selection being possible at all. Evolution theory thus explains the emergence of essential forms and substances from the accidental, relieving us of attributing the order of things to an form-giving telos or origin. It simply turns the terminological framework of world-description upside-down. Evolution theory is not a theory of progress, and it does not deliver projections or interpretations of the future. Autopoiesis, as outlined above, enforces a revision of the concept of "adaptation". Adaptation is a condition, not the goal or outcome of evolution: on the basis of being adapted it is possible to produce more and more risky ways of non-adaptation - as long as the continuation of autopoiesis is guaranteed.

The three separated processual components of evolution can be related to the components of society, conceived as a communicative / social system:

- *Variation* varies the *elements* of the systems, i.e. *communications*. Mainly variation means deviating, unexpected, surprising communication. It may simply be questioning or rejecting expectations of meaning. Variation produces raw material and enables further communication with more open connections than before.

- *Selection* relates to the *structures* of the system, i.e. the creation and use of *expectations* that control communication. Positive selection means the choice of meaningful relations that promise a value for building or stabilizing structures. Selections serve as filters to control the diffusion of variations. Religion has been such a filter. Truth, money, power, as symbolically generalized media serve as filters in modern societies.

- *Re-stabilization* refers to the state of the evolving *system* after a positive / negative selection. It has to take care of the *system-compatibility* of the selection. Even negative selections have to be re-stabilized, because they remain in the system's memory. Today stability itself becomes a more and more dynamic concept, indirectly serving as a trigger for variation.

Variation, selection and re-stabilization can be related to the empirical reality of evolving systems, thus allowing its re-interpretation in the light of evolution theory. For example:

- *Early segmented* societies (families, clans, …), where communication mainly happens as interaction between people present, hardly need the distinction of variation and selection, because every interaction is aiming at immediate acceptance or refusal.

- *Stratified hierarchical* societies have problems to differentiate between selection and re-stabilization, because the main criterion for selection is stability.

- *Modern differentiated* societies differentiate variation / selection as well as selection / re-stabilization, but have problems to distinguish re-stabilization and variation, because stability is of extremely dynamic character and provides the trigger of evolutionary variation. Here we may identify designing, the deliberate, purposeful creation of variety, as a constituent of modernity.

6. THE PROBLEMS OF CONTROL AND PREDICTION – CAUSALITY GAPS

The previous findings allow us to summarize as follows: Designing consists of interacting and co-evolving autopoietic systems and artefacts. Random mutations in nature plus deliberate decisions and accidental events and connections in social life initiate open-ended processes of self-organization, in which positive and negative feedback interact and produce changing patterns that may at some point assume relatively stable forms, called fashions or trends. This kind of mutual causality implies, that it is not possible to exert unilateral control over any set of variables; interventions are likely to reverberate throughout the whole. Though it is often possible to spot an initial "kick" that sets a system moving in a particular direction, it is important to realize, that to our understanding such kicks are not the cause of the end result. They merely trigger transformations embedded in the logic of the systems involved.

We can identify two problem areas: (1) *control*, due to the system / environment distinction, and (2) *prediction*, due to the variation / selection / re-stabilization distinction.

(1) The problem of control:

Luhmann' systems theory provides a map of the possible gaps related to these interventions, called design (Fig. 3). Artefacts as artefacts are assumed to function; this is not the primary task of designing. With respect to the autopoietic systems, I introduce the following gaps, which are always occurring in interaction with different shares, according to the specific design task:

- *organisms* → *the "function gap"*, which indicates, that it is not a trivial (…) task to adapt an artefact to an organism, for example, because bodies cannot speak…

- *consciousnesses* → *the "taste gap"*, which indicates, that it is not a trivial (…) task, to coordinate individual consciousnesses, for example to optimise a solution for the 80 million consumers of the German market. They are all different, and they cannot speak about their taste in clear and distinct manner…

- *communications* → *the "fashion gap"*, which indicates, that it is not a trivial (…) task to generalize a variety of information gathered from individual consciousnesses and to transfer this into the shape of an artefact, for example to plan a new collection of household goods for the Turkish market…

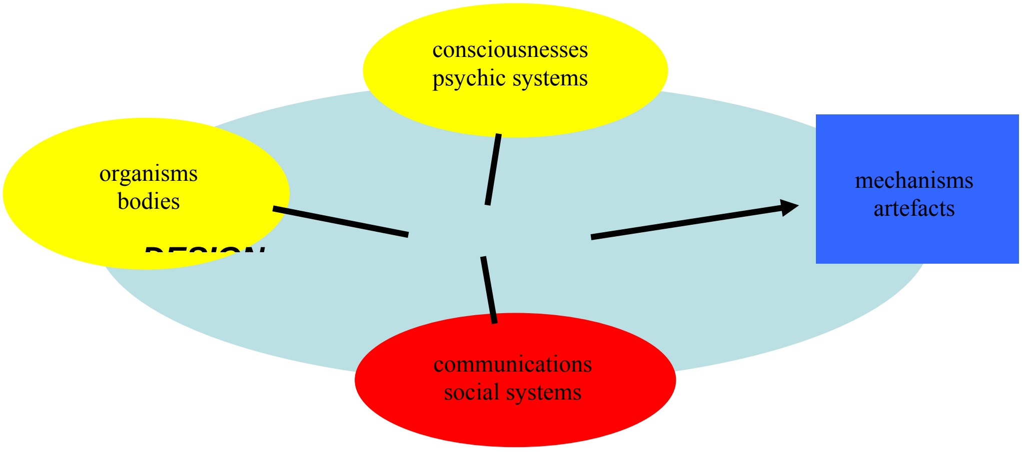


Figure 3. The "scandal of split causality", 3 autopoietic systems + design (Baecker 2000).

(2) The problem of prediction:

- *Variation* is aiming at alternatives. This is no problem, since consciousnesses provide abundant "creativity", which is essential for increasing the variety of choice. This is the "timeless" task of *designing artefacts*…

- *Selection* is aiming at the fit of alternatives into structures. This is a problem indeed, because communicative structures are detectable, but not their future stability. Single aspects can be tackled by isolated approaches: organism - artefact gaps by means of ergonomics, consciousness - artefact gaps by means of cognitive ergonomics, communication – artefact gaps by means of market research, etc. So, to a certain degree, *design research* can examine *existing structures*…

- *Re-stabilization* is aiming at the integration of selected alternatives into the system. There is hardly any predictability, because this is a question of long-term viability of selected alternatives within communicative systems. *Futures studies* and *scenario planning* are dealing with *evolving systems*…

Returning to design: The present does not at all mark the "wave front" of progress, but merely consists of what has remained from the past. And so it happens, that we do not live in the best of all possible worlds. Harmony, if at all, is "post-stabilized" harmony, created in our narratives. The study of failed innovations ("floppology") might be a promising approach to improve designing: the "dark side" of the field is probably much richer than the "best practice" view. Design activities happen "in-between", they intervene into the relations of co-evolving autopoietic systems by means of creating artefacts that pretend to improve those relations. The basic problem is neither lacking individual creativity nor insufficient planning, but the *uncontrollable* and *unpredictable* nature of communication in the environment of the artefacts. The most developed instrument for bridging this kind of causality gaps between psychic systems is *language*, which enables communication. Functioning communication is highly improbable. Functioning design is even more improbable…

7. CONCLUSION

To sum up: there are two basic problems related to systemic gaps:

(1) The gaps between *autopoietic* systems involved in designing. This is fundamental systemic "obstinacy", which is labelled or covered with the nice and common, but fuzzy terms "creativity", "subjectivity", "values", "trends", …

(2) The gaps between the *evolutionary* mechanisms involved in designing. Or: the future orientation of design activities. The artefact, once released, remains as it is. The environments of the artefact change in manners, which are *in principle* unpredictable.

At this point I have reached the limits of my argument. Even a perfect language could only bridge one single gap: the interface between a thought, which is an element of a consciousness, and the communicative offer produced by this psychic system. And this kind of ideal language would have to be a private language, which would probably fail with the addressee. A functioning language has to be a deficient compromise, *a medium*. And *design is a medium as well*, but a considerably less universal one compared to language. Language is deficient, but nevertheless optimal. *Language is the problem and the solution.*

The perspective for design research seems to be: To find procedural / practical approaches to deal with the *unpredictability* of the behaviours of interacting *autopoietic* systems. In evolutionary terms, this means a shift from *prediction & control* towards *learning and design*, or: a shift from efforts of *adaptation* towards strategies of *exaptation*, which means the development of stocks of alternatives for meeting unpredictable situations in the future.

In other words: the choice of process models and methods does not matter, as long as you believe in their projective power and convince others of it. Complexity science may be helpful here, but is not at all to be considered as a panacea. *Design is too complex for complexity science!* Maybe now we have a better idea, why "designing for people" (Jones 1970) or even "for the real world", is so difficult: The entire real world is complex!

REFERENCES

Alexander, Christopher (1964) Notes on the Synthesis of Form, Cambridge, Mass., Harvard University Press

Baecker, Dirk (2000) "Die Theorieform des Systems", in: *Soziale Systeme* 6 (2000) H. 2., Opladen, Verlag Leske + Budrich: 213-236

Bar-Yam, Yaneer (1997) Dynamics of Complex Systems, Boulder, Colorado, Westview Press

Campbell, Donald T. (1960) "Blind Variation and Selective Retention in Creative Thought as in other Knowledge Processes", in: *Psychological Review* 67 (1960), pp 380-400

Cross, Nigel (2001) "Designerly Ways of Knowing: Design Discipline Versus Design Science", in: Design Issues Volume 17, Number 3 Summer 2001: 49-55

Dewey, John (1986) *Logic: The Theory of Inquiry*, Carbondale, Southern Illinois University Press

Glanville, Ranulph (1982) "Inside every White Box there are two Black Boxes trying to Get out", in: *Behavioral Science* 27, 1982: 1-11

Graham, Stephen; Marvin, Simon (1996) *Telecommunications and the City. Electronic spaces, urban places*, London, Routledge

Horgan, John (1995) "From Complexity to Perplexity. Trends in Complexity Studies", in: *Scientific American*, June 1995

Hybs, Ivan; Gero, John S. (1992) "An evolutionary process model of design", in: *Design Studies* Vol 13 No 3 July 1992 pp 273-290

Jonas, Wolfgang; Meyer-Veden, Jan (2004) *Mind the gap! – on knowing and not-knowing in design*, Bremen, Hauschild Verlag

Jones, John Christopher (1970, 1992) *Design Methods. Seeds of human futures*, London, John Wiley & Sons, second edition, Van Nostrand Reinhold, New York 1992

Luhmann, Niklas (1984), *Soziale Systeme*, Frankfurt / Main, Suhrkamp

Luhmann, Niklas (1997), *Die Gesellschaft der Gesellschaft*, Frankfurt / Main, Suhrkamp

Maturana, Humberto R. (1985) *Erkennen: Die Organisation und Verkörperung von Wirklichkeit*, Braunschweig, Vieweg

Mikulecky, D.C. (2003) "Definition of complexity", http://www.vcu.edu/complex/ON%20COMPLEXITY.html (accessed 28 Nov. 2003)

Miller, George A.; Galanter, E.; Pribram, K. (1960) *Plans and Structure of Behaviour,* Harvard Center for Cognitive Studies, New York

Miller, Arthur I. (1996) *Insights of Genius. Imagery and Creativity in Science and Art,* New York, Copernicus
Morgan, Gareth (1986) *Images of Organization*, Newbury Park, London, New Delhi, Sage Publications
Riedl, Rupert (2000) *Strukturen der Komplexität. Eine Morphologie des Erkennens und Erklärens*, Berlin Heidelberg New York, Springer
Rittel, Horst W. J. (1971/72) "Zur Planungskrise: Systemanalyse der 'ersten und zweiten Generation'", in: Ders. *Planen, Entwerfen, Design* 1992 S. 37-58 (Original 1971/72)
Rittel, Horst W. J. (1972) "Second-generation Design Methods", in: Cross, Nigel (ed.) *Developments in Design Methodology*, John Wiley, Chichester 1984 pp 317-327 (Original 1972)
N.F.M. Roozenburg, J. Eekels, (1991) *Product Design: Fundamentals and Methods*, Chichester, Wiley
Simon, Herbert A. (1969, 1981, 1996) The Sciences of the Artificial, Cambridge, Mass., MIT Press
Weaver, Warren (1948) "Science and Complexity" in: *American Scientist* 36 (1948) pp 536-544
Wiener, Norbert (1948) *Cybernetics or control and communication in the animal and the machine*, Cambridge, MIT Press

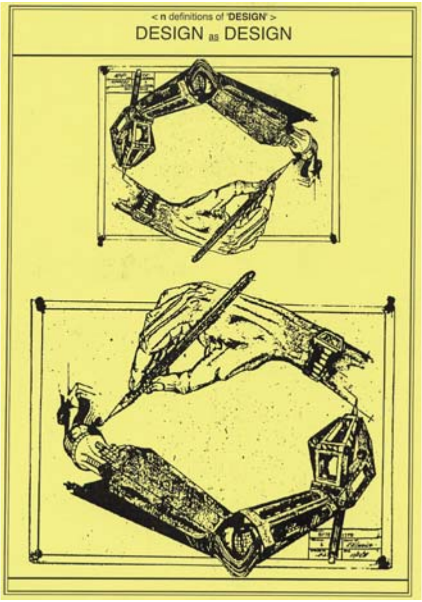
“DESIGN” — Complex definitions

Nec Teymur

Emeritus Prof, METU, Ankara / Visiting Prof., Univ. of Dundee, DoJ College of Art & Design
<n@teymur.co.uk>

ABSTRACT

“Design” is a complex generic term, and is used in a variety of disciplines such as architecture, engineering, industrial design, graphic design, urban design, ...The term itself is a verb denoting an activity; a noun, hence, product(s) and object(s); a practice, hence, a profession and an industry; a mode of graphical representation; and ..... . Moreover, recently, scientists and mathematicians have been discussing whether the God is <u>the</u> “Intelligent Designer”, rather than a “Scientist”, implying that Universe had been ‘designed’ too. From the last definition, several “cosmic / cosmogenic”, “universal” , “theological”, “spiritual”, “teleological”, ... extentions follow. However, in the modern world we are living in, “DESIGN” has attained multiple definitions of quite a different order. This paper, and the accompanying exhibition, presents these in a brief, visual and *designerly* fashion. The key aspect is represented by the following illustration — supported by several other dimensions and definitions:



Keywords: design, complex definitions, disciplines, practices

1. **“TO DESIGN” — as a VERB:** For ex., “I have designed it in 2 weeks”; “ It could have been designed much better”; ...
2. **“DESIGN” as a NOUN:** For ex., “This design”,; “That design’, “Good design”; “Latest design”; ...

DESIGN as VERB

She designed all of it by herself

I have designed it in 2 weeks

It has already been designed ...

Was it designed? — I can't believe it!

I can design it for you

It could have been designed much better than

DESIGN as 'NOUN'

This design

That design

Good design

Bad design

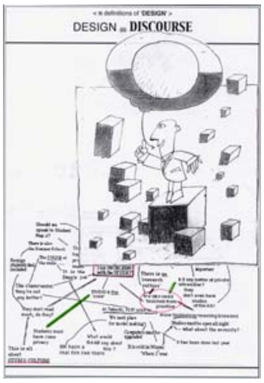
A very successful design

My design

latest design

in-cre-di-ble design

3. **DESIGN AS DISCOURSE:** For ex., “Designers must be trained to do justice to the complexity of the social and industrial variables”



4. **“DESIGN” AS A LABEL:** “Designer Jeans”; “Designer glasses”, “Designer Interiors”; ...

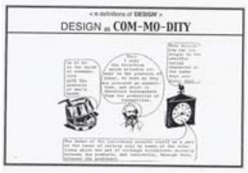
5. **DESIGN AS A MYTH:** “Primitive Hut was the first designed building.”

6. **DESIGN AS PROBLEM-SOLVING:** "We can't solve problems by using the same kind of thinking we used when we created them" [A. Einstein]

also

7. **DESIGN AS A ?UESTION / PROBLEM** to be inquired about, and questioned.

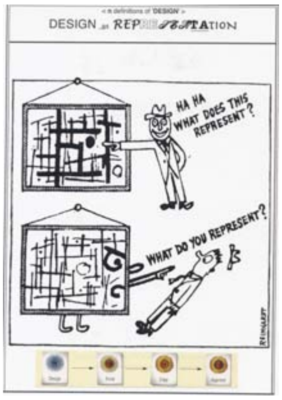
8. **DESIGN AS A OBJECT / THING / COMMODITY:**



9. **DESIGN involving RESEARCH ACTIVITY:**

10. DESIGN AS A PUZZLE-LIKE PROCESS:

11. DESIGN AS REPRESENTATION:

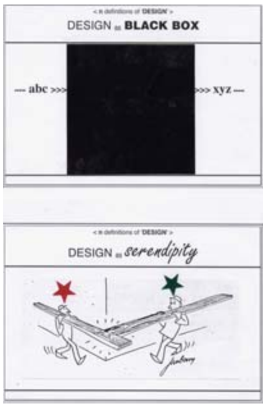


12. DESIGN [partly] an ART FORM:

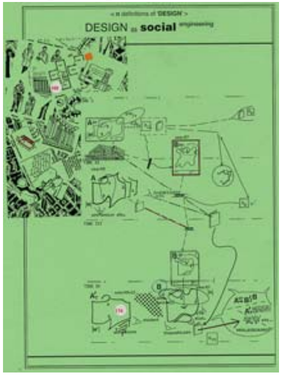
13. DESIGN AS PART OF INDUSTRIAL PROCESS and BUSINESS:

14. **DESIGN AS AN IMAGE-MAKING PRACTICE:**

15. **DESIGN INVOLVING BLACK BOX / SERENDIPITY PROCESSES:**

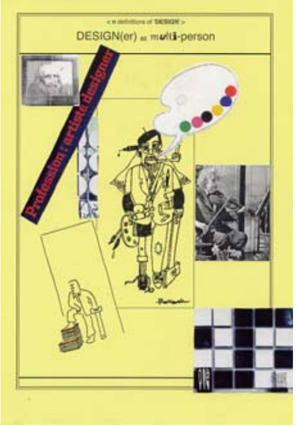


16. **DESIGN[ING] AS SOCIAL ENGINEERING:**



AND, ...

17. DESIGNER as a “MULTI-PERSON”: ideally carrying all these qualities and abilities — art, science, drawing, business skills — if not to the same degree!



ADDITIONAL REFERENCES:

Teymur, N., “Design without Economics:, in Langdon, R. (ed): ***DESIGN POLICY***, London, Design Council . RCA / DRS, 1984, v.3, pp. 75-80.

Teymur, N., Materiality of Design, ***BLOCK READER in VISUAL CULTURE***, London / N York, Routledge, 1996, pp.148-166.

Teymur, N., ***Re:Architecture — Themes & Variations***, 2002, London, Question Press.

Measuring Complexity in a Design Environment

A Thomson[a], B Kumar[b], S Chase[c] and A Duffy[d]

[a & d] Dept of Design, Manufacture and Engineering Management, University of Strathclyde, James Weir Building, 75 Montrose Street, Glasgow, Scotland

[b] Dept of Built and Natural Environment, Glasgow Caledonian University, Glasgow

[c] Dept of Architecture, University of Strathclyde, Glasgow

ABSTRACT

This paper presents details of three separate research projects carried out by members of the Embracing Complexity in Design Scottish Sub-cluster. Each is concerned with the measurement of complexity at various levels of abstraction within three key aspects of the design environment, specifically:

- Complexity within design projects
- Complexity within design teams
- Design complexity within a Computer Aided Design (CAD) Environment

An overview of each project is provided together with details of methodology and findings. The similarity and diversity of approaches is considered and compared with generic lessons, and transferability forming the conclusion of the paper.

Keywords: complexity, measurement, design, computer-aided design

1 INTRODUCTION

Complexity is a term normally used to describe a characteristic, which is hard to define and even harder to quantify precisely. The concept of complexity is not entirely clear (Corning 1998). Complexity means different things to different people with the word 'complexity' being used in many different ways (Perrow 1965, Mohr 1971, Thompson 1981, Waxman 1996 & Hobday, 1998). Researchers have defined it in the context of their respective fields of research (Lewin, 1994). Corning (Corning,1998), identifies properties commonly associated with the term complexity. He states that complexity often implies the following attributes:

- A complex phenomenon consists of many parts
- There are many relationships/interactions among the parts
- The parts produce combined effects that are not easily predicted and may often be novel

It is widely believed that complexity is largely connected to the subjectivity of the observer (Ashby 1973, Lewin 1994, Waxman 1996, Dijkum, 1997, Salingaros 1997 & Corning, 1998).

Some decisions taken at the early design stages often fail to deliver outputs that meet the expectation of customers (Austin, 2002). These failings are attributed to a lack of understanding of complexity of and can result in a number of costly changes and even redesign. It has been suggested that to achieve a better understanding of a project, its complexities should be measured so that fresh approaches can be developed for systematically reducing complexity (Chryssolouris, 1994). A variety of approaches for quantifying the complexity of physical systems exist (Bar- Yam 1997).

2 MEASURING COMPLEXITY IN A DESIGN ENVIRONMENT

The following sections provide an overview of each of the projects introduced in the abstract.

2.1 A Framework for Measuring Complexity of Design Projects

Reported average schedule and cost overruns in design projects range between 41% - 258% and 97% - 151% respectively (Norris,1971 & Murmann,1994). One recent and highly publicised example is the Scottish Parliament Building which overran hugely in both cost and time from original projections. Possible factors in these discrepancies, amongst others, can be attributed to projects being more complex than originally anticipated in its early stages together with poor planning and estimating (Muir 2000). Various researchers have recognised the importance of objectively measuring complexity, as an aid to addressing the cause of such engineering and management related problems (Chryssolouris, 1994, Wiendahl,1994, Baccarini, 1996, Little,1997, Calinescu, 2000 and Frizelle,2000)

Sinha (Sinha et al, 2004) describes a framework for measuring the complexity of a project. Primarily, the framework was developed to support projects within the domain of engineering design. However, it is felt it has a role to play in Project Management in general. The framework generates a Complexity Index for the project with respect to a particular human resource involved in carrying out that project. In essence, the framework provides the Project Manager with a tool which helps identify the possible manifestation of complexity within the project process and the ability to plan accordingly to minimise its impact. The framework consists of five main components or modules as shown in Figure 1:

- **Library of Project Activities –** a library of activities required to carry out a project.
- **Subtask Selection Module –** each project activity identified from the Library of Project Activities consists of a number of subtasks, this module selects the most appropriate subtasks for each particular project activity.
- **Solution Steps Selection Module –** there are a number of methods or 'solution steps' for carrying out each of the subtasks identified by the Subtask Selection Module. The Solution Steps Selection Module allows the most appropriate solution steps to be identified.
- **Information Measuring Module –** this module measures the amount of information content for each of the subtasks through its particular solution steps. The method adopted for measuring information content is based on Shannon's Theory.
- **Information Processing Module –** consists of two sub components, the Library of Complexity Justifying Factors and the Complexity Index Generator.

  Library of Complexity Justifying Factors **-** the aim of this library is to facilitate the Project Manager in identifying reasons to justify a CGF with regard to the dimensions of a project activity. Complexity Justifying Factors are classified as primary or secondary.

  Complexity Index Generator **–** the output of this module is a Partial Complexity Index (PCI) which is generated for each project activity. PCI's are summed to give the Complexity Index for the project. Information Processing Scales are used to classify the project activity (and finally the whole project) as simple, medium complex or extremely complex. This is accomplished based on defining some thresholds of the total amount of information processed in executing an activity and, thereby, the project after summing this up for all the activities.

The method used to establish a quantitative measure of complexity in this research is based upon Shannon's theory of information (Shannon 1948). The concept of entropy has been used to measure the information content of a project. Information content has been considered previously in establishing a measure of complexity (Suh, 1990, Frizelle 1995, Basem et. al 1999, Calinescu et.al 2000, Frizelle et. al 2000)

Generation of the complexity index is a highly interactive process with the project manager and is therefore "context dependent" based on the experience and subjective views of the project manager. The Complexity Index is generated based on the total information content "Ic" for a project activity. Information Content of a project activity is measured based on a development of Shannon's equation of entropy (Shannon, 1948) for defining the expected amount of information necessary to describe a system

$$I_c = -\sum_{i=1}^{S} \sum_{j=1}^{T} p_{ij} \log_2 p_{ij} \text{ --------------------------------------(1)}$$

Where

S= number of sub tasks of a project activity

T= number of solution steps of a sub task

p_{ij} = probability of a sub task 'i' to be in solution step 'j' at a particular instant of time

Complexity Generating Factors (CGFs) are also taken in to account as the causes of complexity within the different dimensions of a project activity. A number of stages are undertaken, specifically:

Stage 1 – Initially the Project Manager will determine what activities the particular project consists of. This is supported by the Library of Project Activities which allows the Project Manager to select project activities archived in the library or if appropriate input activities.

Stage 2 – This next activity is to determine the subtasks that are encapsulated within each project activity identified in stage 1. The project manager determines the level of granularity that is required here. The Subtask Selection Module can provide assistance in highlighting common subtasks from previous projects.

Stage 3 – For each subtask identified in stage 2 the method for its execution is identified. This is known as the 'Solution Steps' for the subtask. This stage requires the number and nature of solution steps to be defined together with their estimated completion times, the summation of which determines the estimated duration of the subtask. Also at this stage the framework will facilitate the Project Manager in identifying probable causes of complexity termed as 'Complexity Generating Factors' from the CGF Matrix a tool which helps the Project Manager to consider the common complexity triggers termed in the framework as CGFs thus identifying areas which may be a cause for complexity in a particular project activity.

Stage 4 – This stage involves the computation of the information content of the project activity. This stage essentially consists of applying the equation 2 to initially find the information content of the activity in question and then contextualise the value obtained. Eventually, a summation of these (partial) complexity indices gives the CI of the project. Philosophically, the information content for an activity is the summation of the contribution each of its sub-tasks makes in accomplishing that activity. Equation 2 effectively measures this contribution of each of the sub-tasks as the time it takes to execute that sub-task. It is clear that equation is solely dependent on time. This, of course, makes sense in light of its origins being in Shannon's information theory which effectively says that the longer it takes to communicate a piece of information the higher the amount of information transmitted between the communicator and the receiver of the information. The final step in this stage is to assign values in the information processing scales. The presented framework validated well against realistic scenarios through its use in case study projects and comparison with the project manager's subjective views on the complexity of activities in the project.

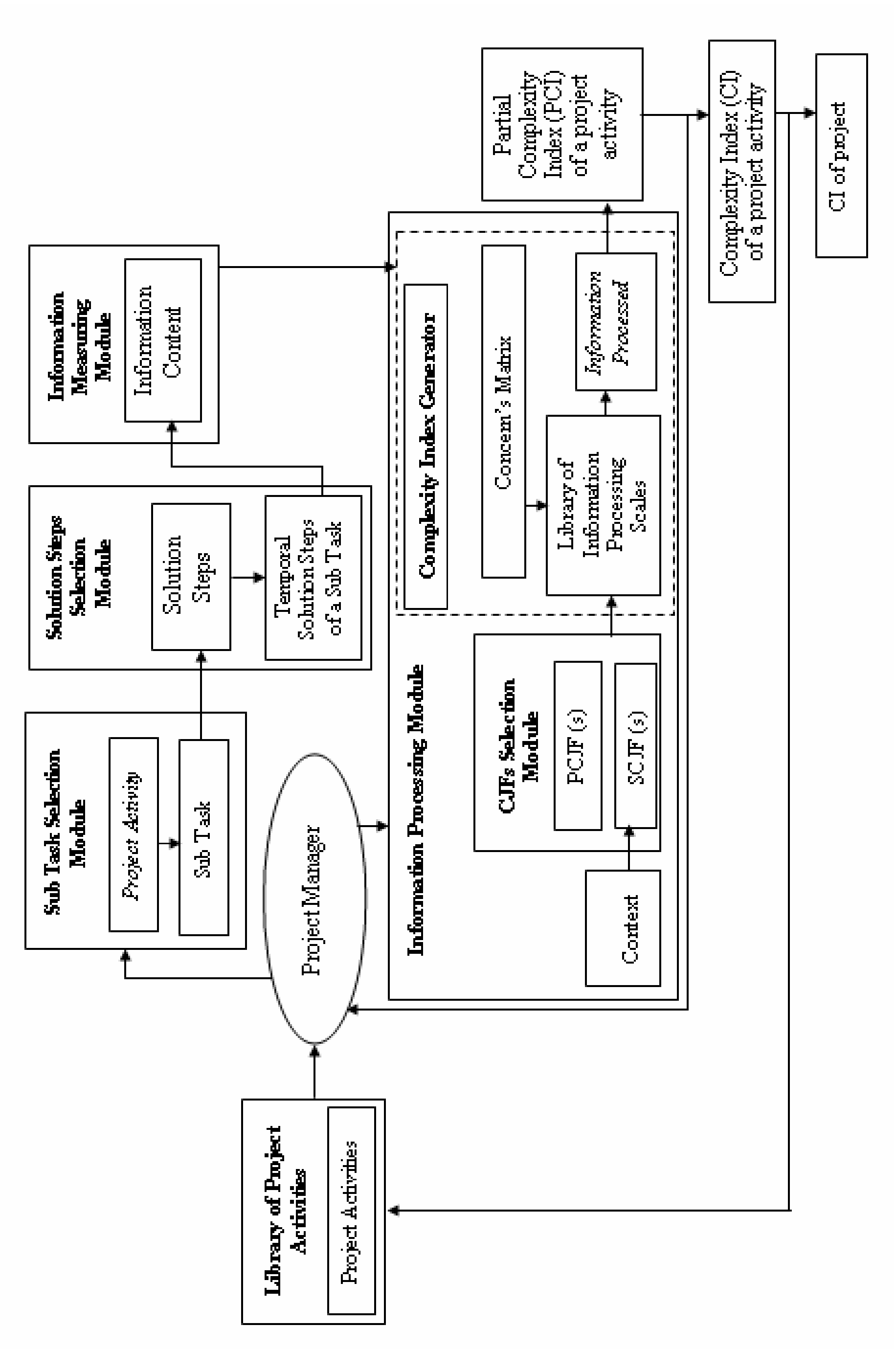


Figure 1. Framework for Measuring Complexity

Figure 2. Design Complexity Map

2.2 Design Complexity Map - Complexity of Teams within the Product Development Process

Duffy et al (Duffy et al, 1995) have identified the factors and issues of design complexity and collated them into the Design Complexity Map shown in Figure 2. This map details the main factors and issues that are considered to influence complexity in design within a team environment. The main factors are described as:

- The artefact being designed
- The design activity itself
- The actors involved
- The decision making process
- The considerations impinging upon design
- The knowledge and sources used and generated

The Artefact – over the decades the marketplace has been increasingly expanded to include more complex and sophisticated products. Product sophistication can be attributed to a number of issues which influence complexity, some of which can be identified from manufactured artefacts. For example the complexities attributed to multi-components, the mixture of technological disciplines, the geometrical complexity of the product, etc. There also exist other issues that influence design complexity and are only visible during design, rather than the resulted manufactured product. Among these are dynamic, multi-staged, multi-functional, multi-faceted and alternative issues.

The Design Activity - Design is a complex activity. It is an activity whose nature has been described by various classification types which focus on issues such as the existence and/or amount of 'innovation' employed during design. For example original, adaptive, variant, routine, innovative, creative, incremental, radical, modular or architectural, repeat order, variant, innovative or strategic. It consists of a number of closely related phases which progressively focus on more detailed levels of the design solution and within which design solutions synthesised, analysed and evaluated. Such issues of complexity can be classified into those related to types and stages of the design activity. Further issues include iteration, uncertainty, distribution , interdependencies and sequences.

The Actors – Actors are used to carry out design. These are entities which can be brought to bear on the design activity and which can be employed to facilitate design. The complexity imposed by actors originate from different types of actors, the structures of human actors, and the dynamics involved in actors. This work proposes two basic types of actors these are humans and tools.

The Decision Making Process – The process of decision making can be typified by highly dependant, conflicting decisions. Such decisions are typically multi-dimensional and as such the decision making process involves the effective and efficient selection of that which is optimal to the particular situation. The design process itself can be considered as a process of decision making. A number of aspects of decision making in the context of design contribute to complexity these being types, time span, distribution, uncertainty and dynamics

The Considerations Impinging Upon Design – The considerations referred to here involve those which are internal and external to the enterprise/company carrying out design

- Internal: this perspective acknowledges that design is not an isolated process within a company. Rather it is part of a higher level , more complex process i.e. the business process and as such should be carried out with reference to the overall business plan. Further internal considerations are related to the necessary acknowledgement of a life phase system from planning, fabrication, assembly through to recycling and deposition and awareness of the company in-house technology and rules and regulations
- External: the design process is influenced by a number of considerations which are external to the company perspective. This alternative perspective represents the world external to the company and involves politics, legislation, the environment, competitors, technology and customers.

The Knowledge and Sources – The initiation, continuation and successful completion of the development of an artefact is dependant on a wide range of knowledge, information and data. A number of issues contribute to the complexity of such knowledge and associated knowledge sources under the heading of characteristics, types, sources and forms and origin of knowledge.

The objective of this work is not to identify a "numeric" measure of complexity rather the factors and issues influencing and contributing to it. With a view to optimising design performance and hence design productivity.

2.3 Measuring Design Complexity within a CAD Environment

Although there is an abundance of published literature on complexity (Garey & Johnson 1979; Shannon & Weaver 1963; Simon 1996) much is concerned with complexity in nature and relatively little is relevant to CAD. CAD complexity is defined (Chase, 1999 & 2002, Murty,1999) as being dependant upon two key factors:

- Design Complexity: the appearance of the object to be modelled
- CAD Complexity: the actual CAD embodiment of the design

Furthermore three sources of CAD complexity are distinguished:

- CAD data – information content of the CAD model
- CAD structure – associated with the models file organisation
- Application Software – associated with application software functionality

During this research an experiment was carried out where seven individuals built CAD models of the same four buildings. Data was extracted from the resulting 28 CAD models and compared using five identified CAD subsystems, namely object differentiation, object grouping, file grouping, application grouping and presets. Within these subsystems measurements were taken on a number of variables these being:

- Blocks v's non-grouped objects v's xrefs
- Layers
- Multiple file usage including xrefs
- Colour styles
- Model file size
- Number of objects
- Number of block definitions and block instances
- Model file size and number of objects after one iteration of block/element explosion

Ratios of these values were used to provide a crude metric of relative complexity.

The ability to evaluate CAD complexity at the beginning of a project has many potential benefits including (Chase & Murty, 1999 & 2002):

- Helping to provide greater understanding of CAD organisation at the project planning stage;
- Providing the perspective for a deeper understanding of CAD model organisation;
- Matching project complexity to knowledge and skill levels more accurately;
- Controlling the modulation of complexity during the course of a project;

3 SUMMARY, CONCLUSIONS AND FUTURE WORK

Three different methods of evaluating and measuring complexity within the design environment are described. There are clear elements of transferability and development that can be considered for future work:

The comprehensive Design Complexity Map presented in 3.3 could be adopted as part of the Framework for Measuring Complexity of Design Projects. Similarity exists in the fact that Complexity Generating Factors are already considered as part of the Framework for Measuring Complexity of Design Projects, and that the Design Complexity Map provides a comprehensive overview of the factors affecting complexity in a design team environment.

It would be interesting to apply the adaptation of Shannon's equation of entropy adopted in the Framework for Measuring Complexity of Design Projects described in 3.1 to measuring complexity within a CAD environment using the same case studies described in 3.2. This would allow a direct comparison of the accuracy of results using both approaches.

The framework for measuring complexity described in 3.1 does not cope well in situations with high degrees of dependencies and further work is required to handle such scenarios. Furthermore, the framework consumes relatively high overheads in terms of processing time required to work out the complexity index.

REFERENCES

1. Ashby, R. W., (1973). Some peculiarities of Complex Systems Cybernetic Medicine No 9 pp1-7.
2. Austin, S., Newton, A., Steele, J., and Waskett, P., (2002) Modelling and managing project complexity International Journal of Project Management Vol 20, No 3 pp-193.
3. Baccarini, D., (1996) The Concept of Project Complexity- a review International Journal of Project Management Vol 14 pp 201-204.
4. Bar-Yam, Y., (1997) Dynamics of Complex Systems, Addison-Wesley (Reading PA).
5. Basem E.L., and Kai Y. (1999), The components of complexity in engineering design, IIE Transactions No 31 (1999) pp 925-934.
6. Calinescu. A., Efstathiou, J., Sivadasan, S., Schirn, J., and Huaccho, H. L., (2000) Complexity in Manufacturing: An Information Theoretic Approach International Conference on Complexity and Complex Systems in Industry, University of Warwick, UK, pp 19-20.
7. Chase S, Murty P, (2002), "Evaluating the Complexity of CAD Models as a Measure for Student Assessment" in Eternity, Infinity and Virtuality in Architecture: Proceedings of ACADIA 2000 Ed M J Clayton, G P Vásquez de Velasco (Association for Computer Aided Design in Architecture) 173-182
8. Chase S, Murty P, (1999), *"Evaluating CAD Complexity"*, *International Journal of DesignComputing* *http://www.arch.usyd.edu.au/kcdc/journal/vol2/dcnet/sub11/index.html*
   Chryssolouris, G., (1994) Measuring complexity in manufacturing systems Working paper Department of Mechanical Engineering and Aeronautics, University of Patras 26110 Greece.
9. Corning, P. A., (1998) Complexity is Just a Word Journal of Technological Forecasting and Social Change Vol 58 In Press, http://www.complexsystems.org/commentaries/jan98.html
10. Dijkum, C. V., (1997) From cybernetics to the science of complexity Kybernetes Vol.26, No. 6/7 pp 725-737.
11. Duffy, A.H.B, Duffy, S.M and Andreasen, M.M. (1995). Using Design Complexities in validating the Design Coordination Framework, Proceedings of the Joint Final Conference of ESPRIT Working Groups CIMMOD and CIMDEV- Cooperation in Manufacturing: CIM at Work Conference, Kaatsheuvel, The Netherlands, 28-30 August 1995, pp16-40
12. Frizelle,G., and Gregory, M.J. (2002) Complexity and the Impact of Introduction new Products International Conference on Complexity and Complex Systems in Industry University of Warwick, UK pp19-20.
13. Garey M R, Johnson D S, (1979) Computers and Intractability : A Guide to the Theory of NP-completeness (W. H. Freeman, San Francisco)
14. Hobday, M. (1998) Complexity, Innovation and Industrial organisation International Journal of Research Policy Vol 26 pp 689-710.
15. Lewin, R., (1994) A Simple Matter of Complexity New Scientist pp 37-41.
16. Little, G., Tuttle, D., Clark, D. E. R., and Corney, J. A., (1997) Feature complexity index Proceedings of Institution of Mechanical Engineers Part C Vol 212 pp 405-412.
17. Mohr, L. B., (1971) Organisational technology and organisational structure, Administrative Science Quarterly.
18. Muir, D., (2000) Why Planning is better with estimation than without *www.spc.ca*
19. Murmann, P. A., (1994) Expected development time reductions in the German Mechanical Industry Journal Product Innovation Management Vol 11 pp 236-252.
20. Murty P, Chase S, Nappa J, (1999), "Evaluating the Complexity of CAD Models in Education and Practice" in CAADRIA '99, Proceedings of the Fourth Conference on Computer Aided Architectural Design Research in Asia Ed J Gu, Z Wei (Shanghai Scientific and Technological Literature Publishing House, Shanghai) 105-114
21. Norris, K. P., (1971) The accuracy of project cost and duration estimates in Industrial R & D Journal of R & D Management No 2 pp 25-36.
22. Perrow, C., (1965) Hospitals, Technology, Structure and Goals Handbook of Organizations. Rand McNally, Chicago, USA.
23. Salingaros, N., (1997) A, Life and Complexity in Architecture from a Thermodynamic Analogy Physics Essays Vol10 pp 165-173
24. Shannon, C. E., (1948) A Mathematical Theory of Communication The Bell System Technical Journal Vol 27 pp 379-423.
25. Shannon C E, Weaver W, 1963 The Mathematical Theory of Communication (University of Illinois Press, Urbana, Ill.)
26. Simon H A, 1996 The Sciences of the Artificial (MIT Press, Cambridge, Mass.)

27. S. Sinha, B. Kumar and A. Thomson. 2004. Activity Complexity based Model of Project Management, Proceedings of In Cite 2004, Malaysia, pages 573-578, ISBN 983-2724-15-5, February 2004.
28. Suh, N. P., (1990)The Principles of Design Oxford University Press.
29. Thompson, P., (1981) Organization and Economics of Construction McGraw- Hill, UK.
30. Waxman, M., (1996) OnProblem Complexity http://www.cpm.mmu.ac.uk/~bruce/waxman.txt
31. Wiendahl, H. P., and Scholtissek, P., (1994)Management and Control of Complexity in Manufacturing Keynote Papers, Annals of the CIRP Vol. 43/2.

Developing an Integrated Model of Designing to Aid Understanding of the Complexity Paradigm in Design Practice

Dr Robert A Young

Associate Dean, Research & Consultancy, School of Design, Northumbria University, Squires Building, Sandyford Road, Newcastle upon Tyne, NE1 8ST, UK
Robert.young@unn.ac.uk:

ABSTRACT

This research is about the development of an integrated model of design that combines descriptions of design content and process in order to improve the design practitioner's ability to navigate complex projects. It seeks to support the work of design practitioners and educators by providing them with a framework to contextualize and understand their work, which is germane to their practice and requisite to contend with increasing levels of complexity found within design contexts. The approach is phenomenological, based on a longitudinal study using a hybrid methodology including a theoretical review of existing models of the design process and reflective design practice studies with quantitative and qualitative assessments of practice outcomes. The result is to adopt the navigational analogy used by previous researchers. To advance the view that content-based models are perceived as more useful to designers than models that describe the process of designing. The conclusion combines both types of model to produce an integrated conceptual framework that enables designers to map out levels of decision making against process activities, providing ability to plot progress within a project and exercise a management overview. The process-based element assists planning and forecasting and the content-based element assists navigation and review.

Keywords: Integrated Model, Complex Design, Process, Content, Context, Conceptual Framework

1. INTRODUCTION

> "The creative act consists in combining previously unrelated structures so that you get more out of the emergent whole than you put in". (Koestler[1])

This research is concerned with the nature of design and further exploration of the future scope of design practice and its role within society. The research follows on from previous work published in the proceedings of the 'Futureground' Design Research Society Conference in Monash 2004[2]. The '3rd Degree' plenary at this conference suggested that; research in design leads away from practice. The chief aim and purpose of the research described in this paper was to achieve the opposite effect to inform and support those who practice and to lead some designers that have perhaps become frustrated with aspects of conventional practice back into a different type or level of design practice. The topic of my presentation is about the development of an integrated model of design that combines a description of different levels of design content with descriptions of the design process in order to improve the design practitioner's ability to navigate complex projects. Therefore the starting point for this research was to revisit the old in order to establish the basis of the new! - conducting a review of models of and for designing using a phenomenological approach. The basis of the approach is the perspective of the design practitioner and their sense-making requirement that design models of utility should increase their capacity to understand and contend with the complexity of design problems that they encounter in their practice.

2. A REVIEW OF EXISTING MODELS OF DESIGN

2.1. Design Methods and Scientific Rationale

Design research has seen the formulation and successive refinement of models of the design process. This has been an essential part of design methods research. There has been an obsession with model development relying upon scientific method where-

> 'Science was seen as the only logical and reliable basis for understanding the physical world, a world assumed to be consistent and stable. It follows that observations in such a world will deliver true facts about its nature free from value based theory. Theory would be derived from observation and accumulate as an increasing body of knowledge'. (Putnum[3])

This is the consequence of the reliance upon scientific method and its accompanying philosophy, as the basis on which early design methods research has been conducted.

Various types of models have been developed over the years. These have proved useful devices for describing and explaining design as a phenomenon of human social interaction. But they have not had a truly instrumental effect on the way designing is carried out, nor have they revealed to those who practice design, what the essential structure (in the scientific sense of discovering laws) of the design process is or should be. The reason for this could be that the use of scientific method as a precedent for developing design methodology was misguided. Kuhn[4] and Popper[5] seriously questioned scientific method in their respective work of this period. Cross *et al*[6] re-evaluated the rationale behind scientific method from an epistemological perspective. They concluded that the basic premise behind scientific method relies on certain assumptions, which are essentially contrary to the nature of design as a human socio-cultural activity.
If we accept that scientific method is contrary to design, then -

> It required that the devices for expressing design as a natural phenomenon with a causal/effect characteristic (i.e. theories and models) underwent a complete re-examination (Cross[6]).

Why is science method contrary to design research methods? A reason why scientific method proved less successful in the development of design methodology was expressed by Sayer in 1984, in reference to the social sciences. He said that many of the systems encountered in the social sciences are open (i.e. they lack consistent causal regularity and effect) and that interpretation and judgement plays a crucial role in our ability to develop new ways of understanding the physical world (Sayer[7]).

Conversely, scientific method has been characterised by its basic 'logical inductive' approach whereby a generalisation about a group of related observations is used as the basis of theory construction. In 1984, Harre[8] examined the inherent problems of logical inductive systems in line with this idea; He showed the various attempts made to overcome such problems. Central to the problem is the premise that there is complete uniformity and consistency of scientific phenomena.

2.2. Appropriate Underlying Values for Design Method

In 1981, Cross-and *others*[6] made a distinction between scientific method and its underlying values. They suggested that it was the underlying values of science; 'rationality, neutrality and universalism', which had appealed to design researchers. The suitability of scientific method as the basis for design methods development was the premise of scientific method, which was itself based on the premise that design was susceptible to systematic description and that such a description would be a germane expression of design activity. Therefore, design was seen to possess an underlying structural logic waiting to be revealed through objective investigation, that the systematic description would then enable designers to utilize new technologies to contend with the increasing complexity of modern design problems. To reveal the flaw in such an argument one must first understand the logic used to promote this position.

Principally the argument would follow four premises, the first of which is that design tasks are becoming more complex as the requirements made of new systems, both physical and operational become ever more demanding. The prevailing logic of science method is strongly inductive as a necessary consequence of its empiricist nature and the reliance on what Harre called the principal of the ‘Uniformity of Nature’.

2.2.1. The Uniformity of Nature

Sayer[7] used the term 'closed system' to describe this concept,

> ‘The social sciences deal with open systems but lack the advantage of equivalents in natural science of having relevant closed systems to draw on. The reason for the openness of social systems is that we can interpret the same material conditions in different ways and learn new ways of responding, - we become different kinds of people. Human actions modify the configuration of systems, thereby violating the extrinsic conditions for closure, - our capacity for learning therefore violates the intrinsic conditions’. (Sayer[7]).

No such argument can be used for design were its very existence depends entirely on man’s own need to instigate change in the physical world.

2.2.2. The Formation and Function of Models

So where does this leave the use of models in design methods research and what implications might this assessment have for the development of models of the design process in the future?

Scientific method has at its heart the development of models of natural phenomena. In 1972, Rivett[9] described the function of a model:

> 'A convenient way of representing the total experience which we possess, of then deducing from that experience whether we are in the presence of pattern and law and, if so, of showing how such patterns and laws can be used to predict the future.'(Rivett[9])

Keywords from his definition are: convenient, represent, experience, deducing, pattern and predict.

2.2.3. Model Functions in Science

From this, a comprehensive list of model functions in science can be compiled:

- simplification of complexity,
- presentation of general principles,
- identification of pattern,
- explanation of natural phenomena, and
- prediction of future events.

However, we must remain aware of the empiricist basis and natural science bias of this definition, particularly in the search for pattern and its obsession with the representation of general principles and laws.

2.2.4. Definition of the Function of a Model in Keeping with Design

Perhaps a more suitable definition of the function of a model in relation to design was that given by Echenique in 1963[10]:

> 'A model is simply a representation of relevant characteristics of a reality - a means of expressing certain characteristics of an object, or system, that exists, existed, or might exist'. (Echenique[10])

Again, this definition makes the point that models are only approximations of real-world phenomena. They select only those aspects of a situation that are considered to be important and ignore the rest. In 1970, Echenique[11] also provided a useful definition of the function of a model:

> 'The main purpose of a model is to provide a simplified and intelligible picture of reality in order to understand it better'. (Echenique[11])

2.2.5. Theoretical Frameworks of Models

This is where the theoretical framework behind the model becomes important, because as Rowe said in 1987[12];

> It is the questions that the model is designed to answer that will determine the selection of relevant variables, antecedent conditions, and so on. (Rowe[12])

These questions are formulated from some theoretical standpoint, and as such are separate from the modelling process. Models are largely logically deduced from a theory about some aspect of the real world the theory itself being logically induced from observation.

2.2.6. The Model Making Process

The model making process itself has a strongly logical positivist approach and is precisely the nature of the model formation process described by Rivett[9]:

- observation
- generalisation
- experience
- validation

2.2.7. The Function of Models of Design Methods

Before exploring the function of models of design methods, a brief examination of the model building process (which is itself a model!) is necessary. In the context of using an inductive process to generate inductive models of the design process, Rivett[9] provided a cautionary note that;

> ‘Objectives cannot be treated separately from the model formation’. (Rivett[9])

Such methods would inevitably generate models along the same lines of reasoning as themselves and therefore impose a system of reasoning not intrinsically central to the activities they model.

2.3. A Model Description of Design

What is an appropriate model description of design? In design, one is not aiming for descriptions of universal regularity but solutions to culturally based problems. In 1984 Broadbent[13] said that design problems are affected by the means which are employed to solve them, in a way that science problems are not. Scientific method has concentrated on the nodes of a systematic approach to design problem-solving, rather than focus on their connections. If you accept some sort of differentiation of the whole process into various stages, then any new model should focus on the nature of the movement between nodes.

2.3.1. Activity Based Design Process Models

During the early period of design research the bias towards scientific method blinded researchers to the importance of the intuitive component in design. They selected only those aspects of design that reflected their own rationalist stance, however, design reality is highly complex. Attempts at generalisation through normative modelling of the design process have never been able to generate a representation of intuitive decision making over all design scenarios. This raises the questions:

- What should design research seek to establish?
- Whether intuitive techniques follow any particular pattern of deployment during the design process and if there are particular types of relationships between intuitive and rational techniques that occur regularly? and
- Under what conditions do they hold true?

Once these questions have been addressed then researchers can consider developing activity based models of the design process that will have the potential for direct application in actual design activity as experienced by designers. Another way of saying this is that models of the design process should be more attentive to design content and context rather than just the process being followed.

2.3.2. Models of the Design Process

In 1991 Oxman's[14] research into experience-based knowledge provided a new designer oriented approach to model development. He proposed a design knowledge hierarchy based on increasing levels of abstraction of previously experienced design situations (both design problems and design solutions). Oxman’s work provided what might be referred to as; a theoretical basis for a knowledge-based dynamic model of design. Understanding how designers make progress from one stage of design to the next provided a greater insight into the process and proved instrumental in enabling designers to improve their own methods and ultimately the quality of their designs.

Also in 1991, Roozenburg and Cross[15] identified two generic model types in design methodology:

- a consensus engineering model and
- an architectural/industrial design model.

The divergence of the models from each other was characterised by a series of dichotomies. See Table 1.:

Table 1. Dichotomies of Design Models

Dichotomies of design models	
Engineering Design	**Architectural/Industrial Design**
prescriptive	descriptive
rational	intuitive
linear (Deterministic)	cyclic
algorithmic (Mechanistic)	heuristic
theoretical	empirical (Practical)
problem focused	solution focused

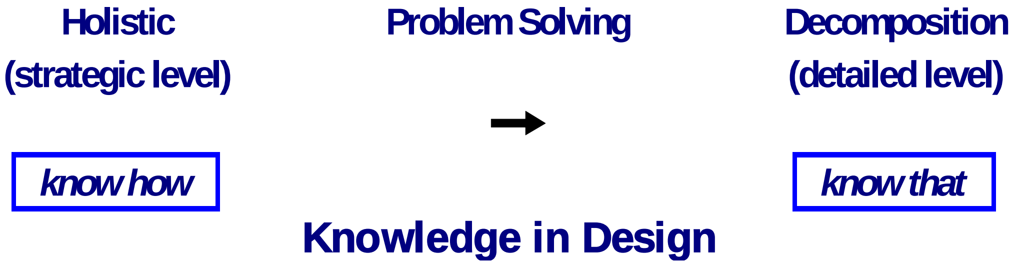
The engineering model remained strongly based on scientific methods, and aimed for value-neutral solutions. This type of knowledge was described as to *'know that'*. Conversely, the architectural model attempted to account for the existing body of experience and intuitive knowledge of the designer. This knowledge was described by Cross *et al*[6] as *'know how'* and, "cannot be made explicit:

> 'It is that tacit knowledge which we know but cannot tell' .

In design, both *'know that'* and *'know how'* knowledge are necessary, but know how, at least according to Cross[6], is central to the activity of design. This follows the original thinking of Polyani[16] concerning types of knowledge.

At the level of detailed design of components associated with the refinement phase of the design process in both architecture and engineering design, Duell[17] in 1983, reasoned that the decomposition of the problem into its constituent sub-problems can be an effective strategy. At the strategic level where the search is for design solutions to complex problems, a near symmetrical process of solution generation and problem exploration occurs. The real difference between the two models lies in the relative balance between these two extremes of design knowledge used at various stages of the process. See Table 2.:

Table2. Knowledge In Design



2.4. The Way Forward

A model should describe a particular balance between the various problem-solving processes in the form of a simplified cognitive map. In 1980, Jones[18] provided a way forward when he defined design methods in the following way.

> 'Design methods are like the navigational tools and charts that the designer uses to plot the course of his journey so as to maintain some control over where he goes'. (Jones[18)]

Using the navigational analogy, both engineering and architectural/industrial design models have concentrated on prescribing new routes or describing existing routes through the design process respectively. Perhaps in the future design model development should focus on the domain in which the journey is to take place and allow designers to plot a course specific to the problems they encounter.

The theoretical critique of existing models of design has highlighted a number of outcomes and observations from research to develop models of the design process:

- Echenique's[11] function of a model; to provide a simplified and intelligible picture of reality in order to understand it better'
- Oxman's[14] research into experience-based knowledge
- Cross's[15] conclusions that *'know that'* and *'know how'* knowledge are necessary to design but 'know how' is central to the activity
- Jones'[18] analogy to design methods as navigational tools to plot a design course and control it.

But how does this critique of existing models fit with the derivation of a content-based model of design from a case study of a complex systems design project? The design of communications consoles for the emergency services was originally used as a complex system design project case study, which was evaluated and reported in Young[19 20]. See Figure 1. below. Results from the case study evaluation sought to refine the process for a specific area of design activity and its analysis revealed combinations of five key factors based on the scale of contribution of design failings to the outcome of the design process – these were:

Five key factors from the analysis
- Communications,
- Knowledge and information,
- Personality, attitude and values,
- Design strategy and policy ; and
- The level of design decision making.

Figure 1. Emergency Service Communications Control Consoles Case Study

3. DERIVATION AND VALIDATION OF THE CONTENT BASED MODEL OF DESIGN

The complexity of the case study meant that, in keeping with other complex projects, there was a hierarchy to the structure of the design problems. Decisions had to be taken at different levels in the hierarchy at different times to progress the project and some important design failings were found to have been caused by difficulties in communicating information about design problems and recognizing their true position in the hierarchy.

3.1. Archer’s Model of Levels of Design

Research revealed that Archer[21] had previously carried out a study of levels of design thinking which showed that:

- Information flow has equal importance to creative solution finding.
- That knowledge elicitation makes information public and therefore undermines power and authority.

He concluded that power and authority are almost exclusively concerned with position in a hierarchy with respect to information. From his research, Archer[21] created a model of levels of design to represent the nature and structure of levels of design thinking. See Table 3.:

Archer’s model of levels of design	
Design at the level of decision	- where the individual designer takes a decision about one small factor in the design task
Design at the level of the product	– is usually the level at which people think and talk about design, where products can be taken as referring to things or systems, which can be designed by the individual designer working alone
Design at the level of the project	– design at this level is the communal activity of the team or organization

Table 3. Archer’s Model of Levels of Design

3.2. A New Model of Levels of Design

From the case study research and the review of analogous research, including Archer’s model, a new model was derived by Young[19] to assist the understanding of the context in which design does and can operate, thereby assisting the practice of designers. This model also recognises three levels of complexity of design practice, but these are named and framed differently by Young[22], namely:

- Design at the level of product configuration and detail, that is - design within a context.
- Design at the level of systems thinking, that is designing context.
- Design at the level of policy formation, - that is design of context (see Figure 2 below).

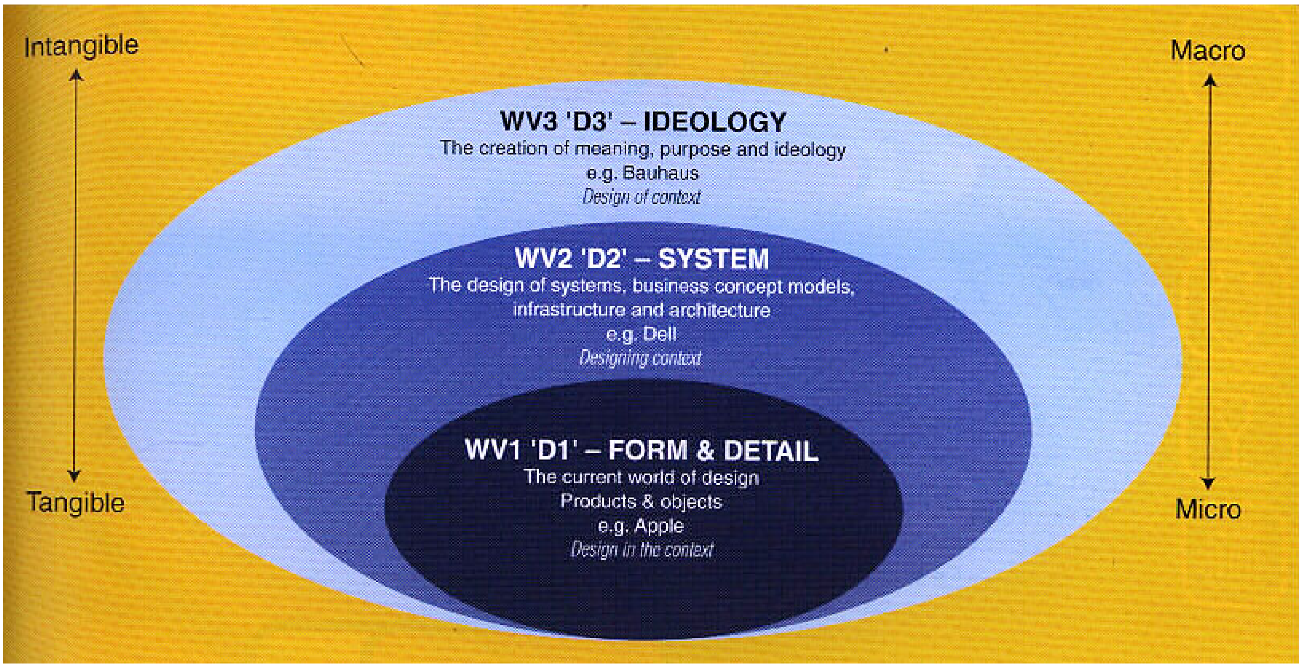


Figure 2. World-view Model of Levels of Design Content

3.3. Correlation of the Model with Other Literature

A recent correlation of the model was carried out with other literature; specifically systems design literature, which has focused on the following work:

- Jones'[23] post design methods work which addresses the complex demands of the contemporary design context and the need for intuition and rationality to co-exist rather than exclude each other.
- Gharajedaghi's[24] 'System thinking: managing chaos and complexity, which advocates a change in mode of thinking to a holistic frame of reference that allows one to focus on relevant issues and avoid the endless search for more detail, through a system framework and methodology, comprising three dimensions of; structure, function and process, and their containing environment, which together define the context.
- Flood and Carson's[25] review of key approaches of systems science to address problems involving complexity using different system models in the technical world and natural sciences and addressing problems in systems that additionally include complexity from human behaviour, learning and cognition as well as complex systemic problem solving approaches.
- Mitchell's[26] Redefining Designing: From Form to Experience, concerning a user-cantered design perspective, focusing on design in terms of human experience rather than physical form and the exploration of collaborative, contextual and intangible design.
- The appeal and real world relativity of Schon's[27] concept of reflective practice for designers, as a mechanism for the growth of their professional knowledge, including; process knowledge or *know how*- and content knowledge or *know that*.
- Popper's[28] model of different worlds; the world of things – of physical objects – the world of subjective experiences – of thought processes - and the world of statements in themselves. He maintained that the distinction between thoughts in the sense of contents or statements in themselves and thoughts in the sense of thought processes belong to two entirely different worlds and it is this distinction that is used in this paper to advocate and promote the utility of a content-based model of design.

The conjecture is that this model creates the potential to attend to the future better than the traditional concerns with products and artefacts. That is, it allows us to design the context rather than to design within the context.

3.4. Educational Application of the Content Based Model

Since the early 1990s, the model has been used in the educational process of successive year groups of industrial design students to assist them to understand the nature of complexity in the design activities implied by the major project design briefs that they devise for their final year undergraduate studies. An action research reflective practice process of enquiry, as advocated by McNiff[29] and McKernan[30], has been used to determine the utility of the model as a learning aid. The process of reflective enquiry see Young[20] has involved the presentation of the concept of levels of design activity to students through a seminar devised for that purpose. Each student's major project brief was analysed in a cross-reference matrix, which contrasted the aims and objectives of the project against the model's structure of levels of design content and activity.

The outcome provides an interpretation of the content and context of the design brief, which is then shared with each student on a personal basis. This allows the student to condone or reject the interpretation as it is presented in the matrix for the project. In this sense the model acts not only as a mechanism to aid understanding of the complexity of structure of design problems but as a mechanism to clarify the mutual understanding of aims and objectives of project work. The feedback loop of the action research as it applies to the use of the model means that the student and tutor have a diagrammatic and powerful medium to focus on as a common point of reference and design content (know that) and process (know how) dialogues can ensue. This application of the model concurs with Jones's way forward of using models of designing as tools to aid navigation of design content and to control the design process. The action research reported in Young[20] shows that design students do use the model to assist their design practice.

3.5. Recent adoptions and adaptations of the model

Recently, the model and has been used to assist the development of a design initiative seeking to promote innovation in design education and practice. The purpose of this initiative is to develop design knowledge and expertise at the levels of system design and the design of policy. A related project called, the Design Innovation Education Centre (DIEC)[31] has grown from the initiative, sponsored by the regional development agency of the North East of England, which used the model to prime attendees at a week long workshop to explore the future of design education and practice, with the aim of supporting the development of expertise in the area of service design (Hollins[32]).

The workshop was led by a London based company, Spirit of Creation, who specialise in the development of service design and who have adopted the content-based model as a core methodology to support their practice. The model has since been adopted by a specialist service design company, Livework as a navigational tool to plan and review their service design projects with industry and to understand the context of their operation.

The emerging nature of the content-based model for design practice has led to the notion that it might be more accurately described as a context-based model to support designing because it enables design issues to be mapped thereby contextualizing their complexity in the decision making hierarchy. However, the researcher was conscious that the content-based model does not allow design issues to be contextualized in this way against the design process that they are part of. In order to do this there needs to be a measure of triangulation between content and process.

The DIEC project set out to build a prescription of the process, methods and techniques to facilitate the development of understanding of the structure of complex service design type projects. The prescription attempted to relate the content model of levels of design practice with a generic model of the service design process. DIEC[31] called this process model their; 'Marketing Design Fusion Model', see Figure 3. below. The derivation of this model by Spirit of Creation has not been described in DIEC[31]. The model attempts to link the disciplines of marketing and design in order that "form and function follows focus and fit with 'real' needs". In keeping with the generic nature of many descriptions of the design process the model recognizes four main stages to a service design project, i.e.: discovery, generation, synthesis, enterprise.

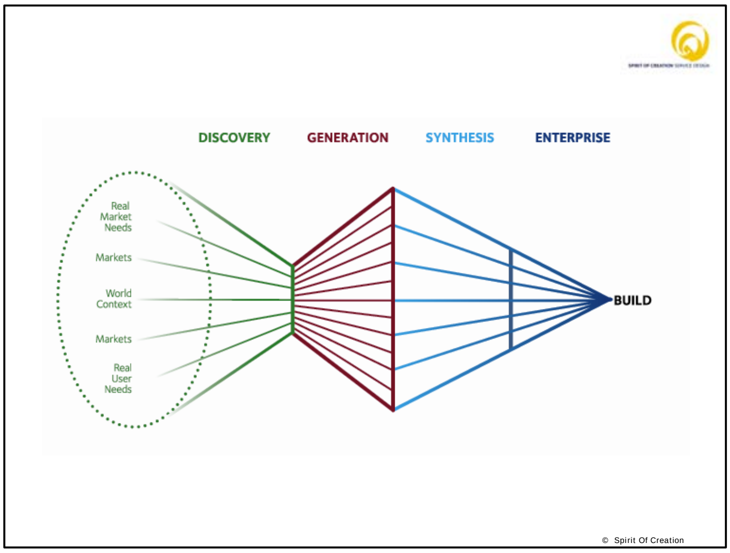


Figure 3. DIEC Project - Marketing Design Fusion Model

When amalgamated, the two models overlap as seen in Figure 4. below. This amalgamated model of design practice is comprehensive in its scope of levels of design practice and the prevailing design process stages that they entail. The model demonstrates that the best designs do and always have begun at the D3 level in order to have a meaningful affect on issues at policy and strategy forming levels of decision-making, despite that fact that clients, commissioners, and customers or users of designs are seldom prepared to give permission to the designer or design team to operate at a level other than D1, i.e.: the level of product detail and configuration. The distinctive proposition of the DIEC project is that is seeks to generate the pre-conditions to enable D2 problems to be identified and for projects to operate at this level of decision making. DIEC's anticipation is that its success as a project will extend the sphere of influence of design because the sphere of the concern of the designer is enlarged; consequently D3 thinking around policy issues will become increasingly permissible for designers and design teams, enabling them to act beyond their existing sphere of influence.

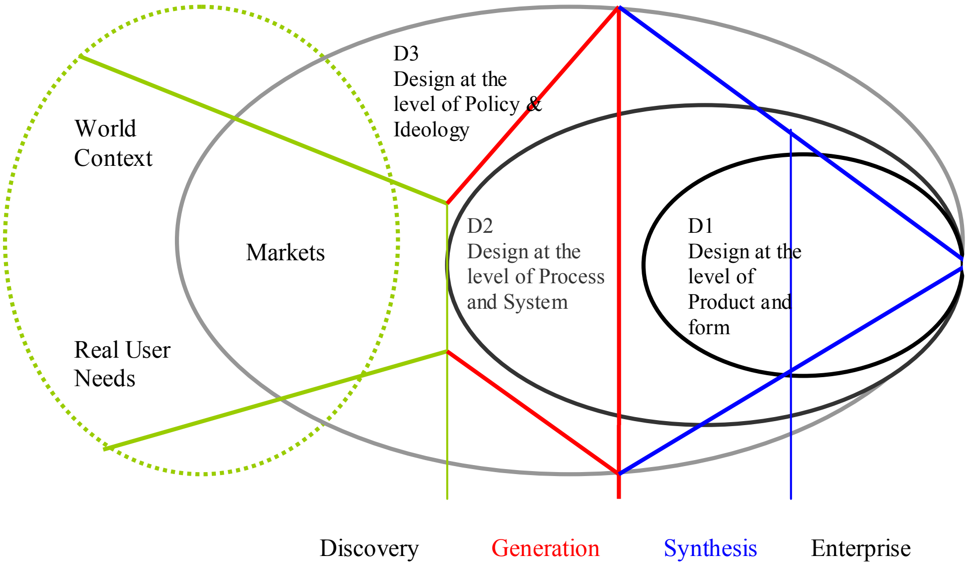


Figure 4. The DIEC Integrated Model of Worldview Levels of Design Content with Market Design Fusion Process

Analysis of the DIEC integrated model shows that it has some measure of effectiveness in being able to raise the awareness of designers to the increasing complexity of design problems that they can encounter. It does this by mapping out the interrelationship of the levels of design content holistically, within a business oriented design process. This goes some way towards a helpful description of universal regularity of the worldview of design for design practitioners. The overlay of the marketing design fusion process also helps to frame the relationship of the levels of problem within discrete business processes or phases. In this sense it follows the trend of product development management models, for example the collaborative and integrated approaches to product development of the State Gate Model (Cooper[33]), Total Design (Hollins and Pugh[34]) and Concurrent Engineering (Roozenburg et. al.[35]). These models see a need to treat separate professional tasks as a whole activity with a unified set of objectives. This is where, for example, the Stage Gate Model appears to be too delineated to ensure effective application for creative problem solving. The guidelines set out by Bruce and Bessant[36] in relation to the application of this type of model suggests the use of cross-functional teams as an essential part of making the design process more holistic and effective.

The authors conjecture is that these models have an inevitable deficiency because they attempt to model a complex, holistic, activity in a two dimensional form and primarily with a linear or sequential narrative. The interactive and cyclical elements of the design process that are the feature of activity- (time and experience) based models are not adequately reflected in these models. This creates a problem of engagement for the designer, for although it might demonstrate Echenique's[11] requirement for a simplified and intelligible picture of reality, their utility stops at the point of being able to use them to map issues and features of the design process within a dynamic professional practice doing/learning context. To paraphrase Schon[27] the 'artistry of practice' pervading every context of the content and process are not represented by these models. In order to do this the author proposes two things. Firstly, that the correct orientation of the process and content elements of the model ought to be arranged as though aligned to x, y Cartesian axes. Secondly, that a third dimension or z axis needs to be added to represent the artistry of practice continuum, or again, as Donald Schön might have said, the reflective design practice practicum. The resulting model is illustrated in Figure 5. below. The model is the subject of on-going evaluation and refinement by the author, in keeping with the reflective practice process that the z axis represents. An international design practice conference is currently being planned for the Autumn of 2007. This conference is currently using the model in order to interpret the nature of design practice of potential contributors and to enable a structure to be created, which makes sense and coherence of the diversity and complexity of their many contexts, content and processes of practice.

This research acknowledges that we can never get a perfect description of the design process. That any working description can only be progressively refined, yet people will still continue to design with, in and around complexity irrespective of attempts to aid understanding and improve effectiveness of practice through model descriptions and frameworks. Although such model(s) will never be perfect, they will also never be totally imperfect. This brings to mind another beautiful expression of this sentiment by St. Paul[37]:

> '…but whether there be prophesies, they shall fail; whether there be tongues, they shall cease; whether there be knowledge, it shall vanish away. For we know in part, and we prophesy in part. But when that which is perfect is come, then that which is in part shall be done away.' (St. Paul[37])

Design in the real world lives with the knowledge that things will never be perfect. Designers appear to enjoy working with this imperfection and uncertainty and see it as an opportunity for creative opportunity!

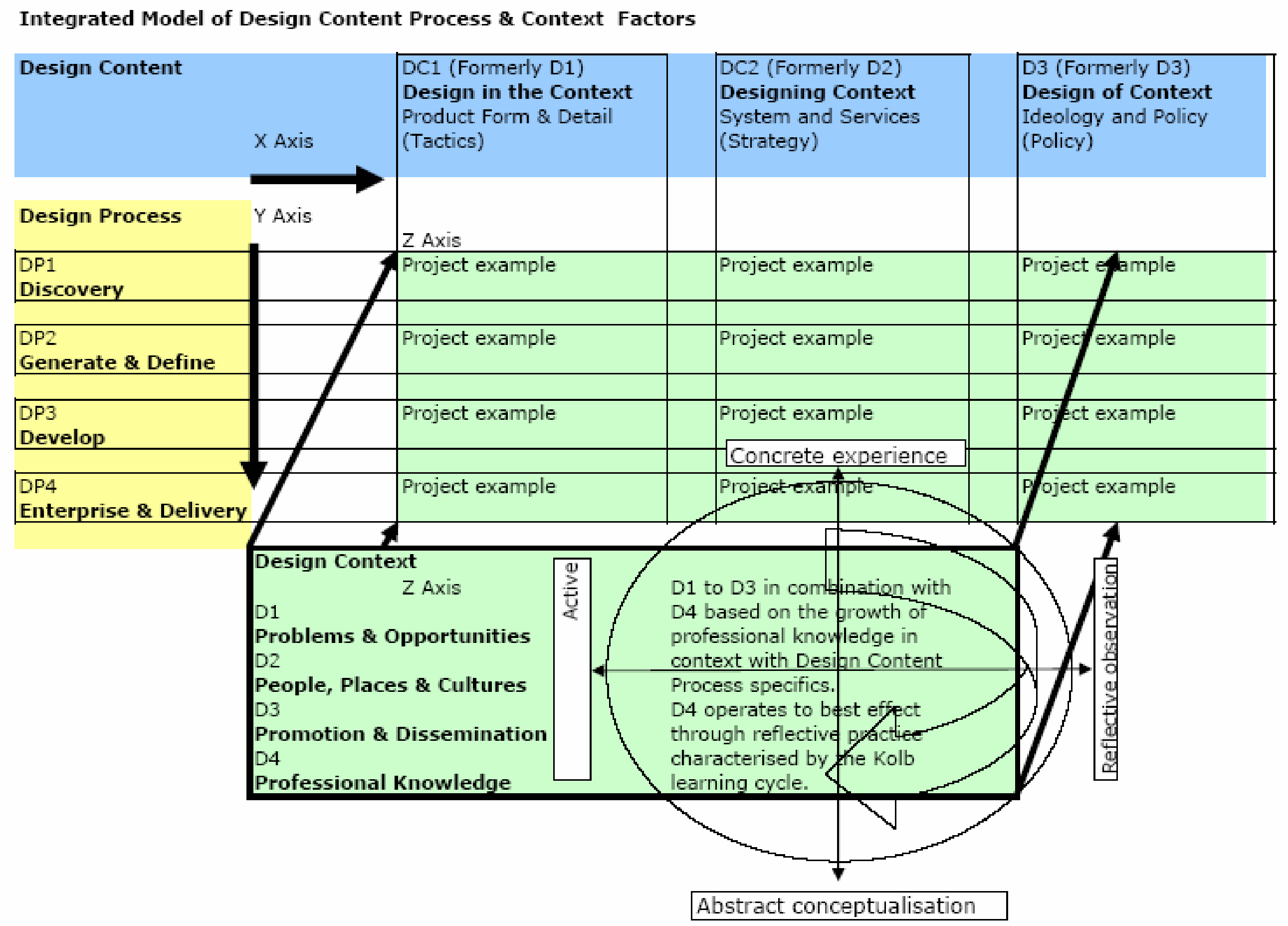


Figure 5. Integrated Context Model of Designing

ACKNOWLEDGMENTS

The author would like to acknowledge the assistance of the ONE North East Design Innovation Education Centre project and Livework, London in the review of the Content based model of design.

REFERENCES

1) Koestler, A. The Act of Creation, London: Hutchinson Cited in Cooper, R., Press, M., *Design Agenda: A guide to successful Design Management,* Chichester: Wiley, 1995.
2) Redmond, J., Durling, D. & de Bono, A. ed., *Futureground, Proc*. Design Research Society International Conference, 2004.
3) Putnam, *Meaning and the Moral Science*, Routledge & Kegan Paul, Boston, pp.129. 1978,
4) Khun, T.S., *The Structure of Scientific Revolutions*, University of Chicago Press, 1962.
5) Popper, K.R., *Conjectures and Refutations*: 3rd ed., Routledge and Kegan Paul, London, 1969.
6) Cross, N. G., Naughton, J. & Walker, D. *Design Method and Science Method,* in Design Studies, vol.2 no.4, pp.195-201, 1981.
7) Sayer, A.., *Method in Social Science, A Realist Approach*, pp. 111-114, 1984.
8) Harre, R, (1983) *An Introduction to the Logic of the Sciences,* pp.3-4
9) Rivett, P., *Principles of Model Building – The Construction of Models for Decision Analysis*, pp. 4, 1972.
10) Echenique, M., Models: A discussion. Working Paper No. 6, *Land Use and Built Form Studies*, University of Cambridge, pp. 1, 1963.
11) Echenique, M., Models: A Discussion, in *Architectural Research and Teaching*, vol. 1, no. 1, 00 25-30, 1970.
12) Rowe, P.G., *Design Thinking*, pp. 163, 1987.
13) Broadbent, G. (1984) Design and Theory Building, in Cross, N. (ed.) *Developments in Design Methodology*, John Wiley & Son, Chichester, UK.

14) Oxman, R. (1990) *Prior Knowledge in Design: A Dynamic Knowledge-based Model of Design and Creativity*, in Design Studies, Vol.11, No.1, pp.17-28.
15) Roozenberg, N.F.M., Cross, N.G., Models of the Design Process. Integrating Across the Disciplines, in Design Studies, vol.12, no. 4, pp. 215-220, 1991.
16) Polanyi, M., *Personal Knowledge: Towards a Post-Critical Philosophy*, Routledge, London, 1973. Originally, Tacit Knowing: its bearing on some problems of philosophy, in, Review of Modern Physics, vol 34. 1962, pp, 601-616.
17) Duell, J., Developing a Detail and Specification, in *Architects Journal*, vol.178, no 31, pp. 45-50 and no 32, pp. 51-60, 1983.
18) Jones, J.C., A Method of Systematic Design in: Cross, N. (Ed) *Developments in Design Methodology*, Chichester: Wiley, pp. 9-31, 1984.
19) Young, R.A., *A Refinement of the Design Process Through an Investigation of the Factors Which Affected the Design of a Communications Console*, PhD Thesis, Newcastle upon Tyne Polytechnic & Institute of Advanced Architectural Studies, University of York, 1989.
20) Young, R.A., Developing Models of Designing to Enable Practice, in *Futureground,* Redmond, J., Durling, D. & de Bono, A. ed., *Proc*. Design Research Society International Conference, 2004.
21) Archer, B.L., 'How Designers Design', Internal Paper, Department of Design research, Royal College of Art, London 1984. Described In: Magee. K., Ed. '*The Elicitation of Knowledge from Designers*', In: Design Studies. Vol 8. No 2. pp 62-69, 1987.
22) Young R.A., Cooper A.M., Blair N.S.P., 'Re-designing Design Education', Keynote Address In: *Emerging Design Paradigm*, ICSID Educational Seminar Publication, Seongnam, Korea, 2001.
23) Jones, J.C., *Design Methods* (2nd edn), New York: Van Nostrand Reinhold, 1992.
24) Gharajedaghi, J., System Thinking: Managing Chaos and Complexity: *A Platform for Designing Business Architecture* Washington, D.C.: Library of Congress, 1999.
25) Flood, R.L., Carson, E.R., *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, Plenum Publishing' New York, 1993.
26) Mitchell, T., *Redefining Designing: From Form to Experience*, New York: Van Nostrand Reinhold, 1993.
27) Schön, Donald A. *The Reflective Practitioner: How Professionals Think in Action*, New York: Basic Books, 1983
28) Popper, K.R., objective Knowledge: An Evolutionary Approach. 3rd ed., Oxford University Press, 1974.
29) McNiff, J. *Action Research: Principles and Practices,* McMillan, London, 1988.
30) McKernan, J. *Curriculum Action Research,* (2nd. ed.) Kogan Page, London, UK, 1994.
31) ONE North East, *Specification for tender: A New Concept in Design-Innovation-Education Centre*, Confidential Report, ONE North East Newcastle, 2003.
32) Hollins, W., *About Service Design*, Design Council, London, 2003.
33) Cooper, R., *Winning at New Products: Accelerating the Process from Idea to Launch*, New York: Perseus, 2001.
34) Hollins, B., and Pugh, S., *Successful Product Development*, London: Butterworth, 1990.
35) Roozenburg, N., and Eekels, J., *Product Design: Fundamentals and Methods*, Chichester: Wiley, 1995.
36) Bessant, J., Why Design? In Bruce, M. and Bessant, J., *Design in Business: Strategic Innovation Through Design*, Essex: Prentice Hall, 2002.
37) St. Paul, 1 Corinthians 13 v8-10, in *New Testament*, King James version, Collins, Glasgow, 1957.

Linking design and complexity: a review

Theodore Zamenopoulos[a] and Katerina Alexiou[b]

[a] Centre for Advanced Spatial Analysis, UCL and Department of Design and Innovation, The Open University, UK
[b] Centre for Advanced Spatial Analysis, UCL and Department of Design and Innovation, The Open University, UK
t.zamenopoulos@ucl.ac.uk

ABSTRACT

The development of a science of design has in reality followed a path parallel to that of complexity science. The paper is an attempt to identify the many ways in which design has embraced complexity (and vice versa) by reviewing fundamental concepts and traditions developed within the two fields. This overview examines challenges and opportunities in the linkage between design and complexity and proposes a route of investigation which focuses on the identification of the design capacity of complex systems.

Keywords: design science and research, complexity science, organizational capacities of systems

1. INTRODUCTION

In attempting to forge links between design and complexity one has to take into consideration that both design science and complexity science have long research traditions complete with a collection of fundamental concepts, methodologies and distinct epistemological assumptions. They are also both very diverse. Taking part in the workshops and conferences organized as part of the activities of the AHRB/EPSRC Embracing Complexity in Design research cluster we have often encountered difficulties in communicating our individual points of view precisely because we each follow different paths in understanding and studying design and complexity. The paper attempts to identify and highlight key ideas and research questions in the two fields and investigate their synthesis. The motivation behind this effort is to help establish a common communication language between researchers on either ends, but also to raise questions and routes of scientific enquiry that can be of mutual benefit.

2. COMPLEXITY RESEARCH

Complexity research has a long and diverse history which can be conceptually traced back to cybernetics, general systems theory, information theory and even game theory. This history has also been built upon much older, but more domain-specific traditions, such as thermodynamics in physics, evolutionary theory in biology, abstract algebra and computability in mathematics and computer science, distributed computing and multi-agent systems in artificial life and artificial intelligence, dialectics and social constructivism in social theory, and last, but not least, holistic approaches in philosophy.

Generally, complexity is seen as a characteristic of a system (and/or its observer) described on the basis of concepts such as size, variety, order and organization. For instance, complexity is defined with relation to the size, or dimensions, of a system, the difficulty to model it, or the resources needed to describe it. Complexity is also identified with variety - diversity and multiplicity - and it is inextricably linked with the concept of entropy. Entropy is a measure of uncertainty or ignorance: it reaches its maximum value when all possible states of a system are equiprobable. Order and organization have a double meaning: they are constraints that a system satisfies, but also represent a process towards a state of minimum entropy. This is typically linked with the critical capacity of a system to self-organise.

The interpretation and understanding of these concepts varies across domains. However, despite the variety of the approaches, complexity is typically coupled with a “constructive” stance in doing science: a disposition to focus on emergence rather than pre-determined order (“let things be constructed”), to use generative methods and simulation instead of decomposition and analysis (“don’t break things apart”), and to embrace holism versus reductionism (“god is in the relations”). As a working position and for the purpose of this review, we will identify this stance with an effort to study systems by preserving their organization, or, to put it differently, to study complexity as a characteristic of the organization of systems. This draws attention to the relational character of systems, the irreducibility of the whole into its parts, and the importance of the context within which a system is situated.

Complexity research is often baffled by three self-referential questions: what is complexity, what is a complex system and what is complexity science and epistemology? There is no intention to give solid answers to these questions in this paper, but only to provide a framework in which answers can be derived. The meaning of complexity and the epistemological and methodological stance that characterize complexity science are the very components of the complexity curriculum and as a result the answers should, and will, evolve together with the science itself. In the following three themes of the complexity research agenda are introduced: the first is concerned with the description of the organization of a system; the second with the origins of organization; and the third with the effects of irreducible organization and the capacities derived by it (see Table 1).

Table 1. Approaches to the study of complexity as a characteristic of the organization of systems

Description of the organization of a system			**Origins** of organization	**Effects** and capacities of organization
Structural	Functional	Behavioural		
Unorganized Size Variety Order-disorder **Organised** Catastrophes Networks Scaling Fractals	**Algorithmic** Size of time/space Min description Randomness	**Deterministic** (sensitivity to initial conditions) **Aggregated** (Order by complex interactions, criticality)	Evolutionary Theory Self-organizing criticality Dissipative structures	Autopoiesis Autocatalysis (M,R) Systems Neural Networks

2.1 The description and modelling of complexity

The first major subject of investigation in complexity research engages in questions such as: What is complexity? Is there a method or measurement that decides whether a system is complex? What is the appropriate language to describe the organizing principles of a system? All these questions are concerned with describing and modelling complexity. It is possible to identify three main, although at times overlapping, directions of research in this area. The first is looking at the structural characteristics of systems, the second is looking at the difficulty of building functional descriptions, and the third defines complexity by looking at the behavioural or dynamical characteristics of systems.

2.1.1 Structural complexity

It is often convenient to think of a system as a set of elements with some sort of structure. In this context, complexity has been (confusingly!) seen both as something identified with an unorganised structure (typically described by entropy measures), or as a characteristic structure which assumes a critical state between order and disorder (typically described by a power law distribution or a fractal like geometry).

Information theory[1] has laid the theoretical background for defining complexity in terms of variety, (lack of) organization and uncertainty. In this context, complexity is the amount of information that a structure is possible to encode. A system with a large number of possible structural configurations (or states) can carry more information than a system with fewer states. Thus, complexity represents the combinatorial capacity of a system. As the combinatorial size of the system increases most improbable events are possible to be described. The intuitive assumption is that an improbable event contains much more information in comparison with a certain one. As a result, complexity is also linked with the probability of a given state or expression of the system. Inspired by entropy quantification in thermodynamics, complexity is defined as a function of the entropy of the system[2].

On the other hand, in biology, physics, and social science, the term complexity has been largely used to describe an emergent/characteristic property in the structure of a system. A predominant example is the characterization of the structure of a system by an exponential function $P(s) \propto s^{-\gamma}$ (Power Law) in the frequency distribution of some quantity s (such as size of components or the number of links that are incident to the components). A power law distribution implies a hierarchy: small sizes are common whereas large instances are extremely rare. This can be compared with other random distributions (such as the Poisson distribution) where such hierarchy is lost. Examples[3,4,5,6,7,8] of such distributions are very common in the context of biological and socio-technical complex networks such as metabolic networks, the world wide web, communication systems, or cities. The observation of self-similar or fractal geometries across scales is another example in the identification and description of complex structures[9].

Finally, in applied and pure mathematics, complexity has been identified with the representation and analysis of hyper structures. This term generally alludes to structures whose components have structures themselves, enabling the representation and study of issues of scaling, variety and organization in a concrete manner. Examples include Q-analysis[10,11], hyper graphs[12] and category theoretic treatments of complexity[13].

2.1.2 Functional complexity

Another convenient way to model a system is to see it as a function that produces an output given an input. Complexity is therefore often defined in terms of how difficult it is to describe the input-output pattern of a system. Roughly speaking, there are two main traditions of research in this area. The first one is a branch of theoretical computer science known as time complexity theory. It is concerned with the resources (in terms of time and space) needed to describe the output of a function given an input of certain size. Complexity is therefore a measure of how long it takes, or how much memory is needed, in order to make this computation. The second tradition is concerned with the compressibility, or the required length, of a program needed to compute a function. It is generally known as algorithmic or Kolmogorov-Chaitin complexity, named after Andrei Kolmogorov and Gregory Chaitin who independently pioneered this approach[14,15,16,17,18]. The length of description of a message cannot be too much more than the message itself. Intuitively, when the length of the description (model) is much shorter than the message itself, then a higher compression is possible and complexity is minimal. Based on this line of thought complexity has been also associated with randomness: in a random message no compressed version of the message is possible and as a result complexity is maximal. In reaction to this last definition, it is possible to argue that a random function can be simulated with a very simple algorithm much like fractals can be defined by a very concise recursive definition. So a random function may be very simple. In this sense, maximum complexity can be situated somewhere between randomness (or non-compressibility) and order. More information on formal measurements can be found in Bennett[20] and Edmonds[21].

2.1.3 Behavioural complexity

Systems are often modelled as dynamical systems. Complexity is then identified by the capacity of a system to exhibit a certain class of behaviours. There are two typical models: mean field models and spatial extended models. In mean field models, only average quantities are represented and spatial correlations are ignored. In this case, differential and recurrent equations are used as fundamental tools to describe complex behaviours. Spatially extended models on the other hand incorporate both temporal and spatial degrees of freedom. In this case, the fundamental tools for describing complexity are lattice or cellular automata, networks and multi-agent systems. It is also interesting to note that in mean field models the concern is with how simple rules can define very complex behaviour that is difficult to predict. On the other hand, in spatially extended systems the interest is more on how highly complex interactions in space and time can lead the system in some form of order.

More specifically, in mean field models recurrent equations of the form $x_{t+1}=f(x_t)$ have been pivotal in studying behaviour in terms of fixed points and patterns related with the stability of a system (when differential equations are used the behaviour is studied in terms of equilibrium points). These equations can exhibit interesting properties such as sensitivity to initial conditions. This is a characteristic behaviour found in a class of systems usually referred to as chaotic systems[2,22,23]. In spatially extended systems, phase transitions are the important characteristics associated with complexity. A simple example of phase transition can be seen in random networks where N represents the number of nodes and M the number of edges. As the number of edges M increases in relation to the number of nodes N, a sudden qualitative change occurs. This qualitative change is the formation of a "giant component", an almost fully connected network[22]. For similar examples of phase transitions see Kauffman[24]. Likewise, the behaviour of complex systems is often characterized by a natural attraction to a critical state[25]. This critical state is described by a power law distribution where small occurrences are very common and large occurrences rare. Other characteristic emergent behaviours that appear particularly in socio-economic systems include phenomena such as segregation and path dependence[26,27,28]. Finally, the behaviour of complex systems has also been classified in the context of cellular automata into fixed periodic, complex, and chaotic[29] – another example where complexity appears in the edge between chaos and order.

2.2 The origins of complexity

The second theme of investigation in complexity research is concerned with the development of theories on the origin and maintenance of complexity. Given the identification of complexity in the previous section the central question here becomes the development of explanatory theories. For instance, given the importance of power law distributions for the survival and functionality of a system, what are the mechanisms and conditions that explain the emergence of such a distribution? There are a few frameworks that dominate the explanations of complexity, some of which existed before but have been amalgamated within the complex systems enquiry like: evolutionary theory[30], game theory[31,32], theory of dissipative structures[33] and self-organizing criticality[34,35].

2.3 The effects of complexity

Up to this point the science of complexity has been identified with the description, modelling and development of explanatory theories of complexity. There are however complementary models and theories that study the consequences of complexity for the functionality of a system rather than the complexity per se. More specifically, in these studies complexity is seen as a cause (rather than an effect) of the emergence of "high-level" functionalities and capacities such as cognition, autonomy, intelligence, language, life, or sociality.

For example, the study of artificial neural networks was one of the very first endeavours to understand cognition and the sophisticated functionalities of the brain, such as memory, pattern recognition, control, or creativity, as an attribute and consequence of the organization of the system. The crux of neural networks in general is that information is distributed between the nodes of the network and processed in parallel as a function of the strength of the links between the nodes. In contrast to symbolic approaches to understanding and modelling cognition, connectionist approaches assume that the high-level functionalities of the brain emerge from the interconnection and dynamic interaction of simple units (neurons) that have the ability to self-organize[36,37,38]. Connectionist approaches are related to other functional or dynamical approaches in understanding and modelling cognitive systems such as those proposed in Port and Van Gelder[39].

Moreover, autonomy[40,41,42,43] and abilities such as communication, language, as well as creation and assignment of meaning[44,45,46,47] have also been studied as abilities derived by complexity. Notably, such approaches take complexity not only as an innate attribute of a system, but also as an attribute obtained by the interrelation of systems with their environments.

The fundamental question of life has too been defined and studied on the basis of organizational properties of complex systems. Auto-catalytic[24,48], autopoietic[49,50] and M-R systems[51] are three very similar examples in which life is explained in terms of self-referential organizing principles and properties derived by the variety and distribution of complex networks. In fact, life and cognition have been in many cases studied together, exactly because they are both considered as high level abilities derived by a similar cause: complexity[52]. Issues of organizational closeness and self-organization have also been associated with social systems[53]. Finally, artificial life represents another very characteristic thread of research which focuses on the effects of organization and complexity for the creation of life. Computer simulations are used as a key vehicle for studying the consequences of emergence and self-organization for the identification and development of life forms[54,55,56]. This research has also fed into the development of behaviour-based robotics[57].

3. DESIGN RESEARCH

Design also has a long standing history as a field of scientific enquiry and has been studied from many different perspectives. As a science it aims to develop theories about how designers think and work, but it also aims to develop methodologies and tools to support designers in their tasks. However, for the purpose of this review we will not treat theoretical, methodological and technological studies separately as the focus will be on how design is perceived and studied. The most commonly quoted definition of design "Everyone designs who devises courses of action aimed at changing existing situations into preferred ones" comes from Simon[58] (pp 111) who also pioneered research in complexity. This definition appreciates design as a natural activity which pervades all professional domains and disciplines. But it also considers design as an emblematic human ability – a characteristic bias in design research. However, while the majority of research is focused on the *designer* and the *design process*, other approaches concentrate on the design *artefact* and its representations. Additionally there is a class of studies that look at design as a *social activity* or as a set of *epistemological* concepts. For a summary of approaches see Table 2.

3.1 Focus on designer and design process

Examples of research in this first case include studying design as problem solving, or as a complex cognitive or knowledge level ability, and investigating methods based on optimization, heuristic search, computational creativity, and so on. One of the first approaches, the so-called design methods research (very much inspired by behaviourist studies of creative human problem solving as well as systemic and cybernetic methodologies), attempted to describe design problem solving abilities by way of a logical structuring of design phases or tasks[59]. In this context, design abilities have been explicitly associated with structured tasks/phases such as analysis, synthesis and evaluation[60,61,62,63,64]. However, it was the seminal work of Newell, Shaw and Simon[65] on human problem solving that paved the way to the development of a cognitive explanation of design systems as information processing systems and, in particular, the understanding of design abilities as search abilities. This view has been an article of faith for a very large number of empirical, theoretical or computationally driven design studies[66,67,68,69].

Studies on cognitive abilities in design have also highlighted the need to incorporate in computational models and tools functionalities such as learning[70,71,72,73], knowledge acquisition, representation and storing[74,75,76], exploration[77,78], associative memory[79,80], constructive memory[81] or analogical reasoning[82,83,84,85,86].

A quite different approach within this first paradigm of studying design is the development of knowledge level formalisations, which concentrate on identifying and modelling the types of knowledge involved in designing[87,88,89].

3.2 Focus on artefact

Design in the second case is seen as a science of artificial objects. It is concerned with possible worlds, their representations, as well as the constraints and choices they incorporate. The focus might be on a wide variety of objects from miniature products to machines to buildings and to cities. Research here can be divided into analytical and generative. In analytical research the focus is on the problem space of design. Examples include studies on the structuring and decomposition of the design problem[90], spatial and morphological analysis of complex objects[91,92,93] and design evaluation. In generative studies the focus is on the solution space of design. Typical examples include description and enumeration of the design space based on grammars and production systems[94-105].

3.3 Focus on social aspects

Focusing on design as a social action draws attention to aspects of communication and collaboration and on the identification of how social structures drive and are driven by design. Schön's criticism[106] of the view of design as problem solving is one of the most seminal examples of design research which focuses on design as an interactive process that takes place within a social environment. Studies of design as a social activity have become the centre of attention in most recent years[107,108,109] and essentially seek to define design abilities in terms of social abilities and interactions. Such studies have been focused both on processes and knowledge structures necessary to support the development of cooperation, as well as the exploration and construction of common knowledge and understanding. Typical areas of investigation include: the study and modelling of mechanisms for communication, collaboration, negotiation, argumentation and social interaction[110-116]; the study and use of external representations and artefacts to contextualize information and extend the cognitive abilities of designers[107,117,118,119]; the study and formalization of design as a capacity of the distribution and organization of socio-technical systems[118,120121,122]; and the identification of knowledge structures that support reflective reasoning and reasoning from multiple viewpoints[87].

3.4 Focus on epistemology

Finally, research on design as epistemology includes developing theories about the nature of design knowledge and establishing design as a domain which is related to – but distinct from – science and art[123,124,125,126,127,128,129]. This is a long-term investigation which covers themes such as defining design as a distinct discipline; identifying the relation between design theory, design research and design science; examining the strengths and limitations of understanding and modelling design systems; and illuminating the nature of design thinking and knowing.

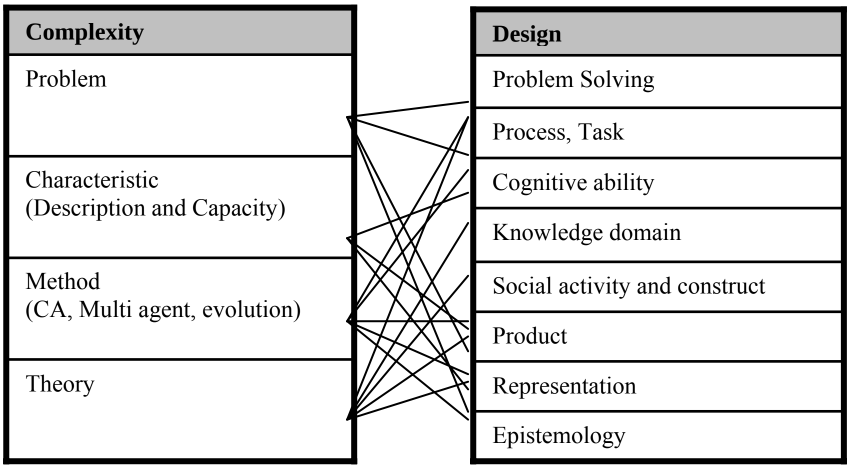
Table 2. Summary of different approaches in design research

Focus on designer and design process	
Problem solving	e.g. Optimization, heuristic search, exploration
Process, task	e.g. Synthesis, analysis, evaluation
Cognitive ability	e.g. Logic reasoning, information processing, creativity
Knowledge domain	e.g. Knowledge level, intelligence, rational action
Focus on artefact	
Product	e.g. Possible, actual and designed objects
Representation	e.g. Representation of structure, behaviour or function
Focus on social aspects	
Social activity and construct	e.g. Communication, collaboration, social norms
Focus on epistemology	
Design science, design knowledge	e.g. Define design in contrast to science and art

4. LINKING DESIGN AND COMPLEXITY

Understandably, the relation between complexity and design has been interpreted in a variety of ways which we can classify here under four different approaches, although there are inevitably many overlaps. Roughly complexity has been perceived 1) as a *problem* encountered in practicing design or understanding and representing design processes and products; 2) as a *characteristic* attribute of design systems and artefacts; 3) as a *methodology* and tool for designing; and 4) as a *theory* for understanding and defining design. A summary of the interconnections between complexity and design are shown in Table 3.

Table 3. Current links between complexity and design



4.1 Complexity as a problem in design

The first approach sees complexity as a critical problem in design (whether it is the process or the product) that we need to manage and reduce. For example, complexity is associated with the difficulty of solving design problems, the combinatorial size of the search space, and the variety of the generated designs. Notably, the complexity of solving design problems is not only because these problems are often intractable, ill-defined or ill-understood, but also because they involve many different participants, with many different goals and needs. Examples of investigation[130,131,132,132,133,134] include studying, measuring and managing the complexity of manufacturing, engineering and construction processes and projects, and looking at problems such as customisation, scheduling, or change management. Undoubtedly, the complexity of processes is tightly linked to the complexity of the product itself or the way we analyze, synthesize and represent it[135,136,137].

4.2 Complexity as a characteristic of design

The second approach sees complexity as a characteristic or attribute of design and suggests design systems can be seen as systems that exhibit complex abilities or have characteristic complex structures. For example design teams are seen and studied as complex networks with characteristic structures and rules of interaction[138,114]. Research here also includes studies which seek to understand and model design artefacts as special instances of complex multilevel systems[139,140] or measure and reproduce unique characteristics of designs – particularly urban forms and patterns[141,142].

4.3 Complexity as a method in design

The third view sees complexity as a set of methods and tools for solving design problems or simulating design processes and structures. For example, methodologies that have now become central in complexity research, such as evolutionary algorithms[143,144,145], or cellular automata and multi-agent systems[146-155], have been regularly used to generate design solutions, solve multi-objective optimization problems or evaluate design solutions, but also to model, represent, visualize and generally support complex design processes and tasks. There is a great tradition in using such techniques to model dynamical processes in cities, simulate the change of urban forms and visualize future planning scenarios (for overviews see Batty, Besussi and Cecchini, and White and Engelen)[141,156,157].

4.4 Complexity as a theory of design

Finally, the fourth approach sees complexity as a theory of design, and as a source of opportunity for advancing and improving design quality. In this sense, complexity can be seen as a set of epistemological concepts that help us approach reality and understand design processes and products: examples[158,159] include concepts such as self-organization, co-evolution, autopoiesis, or anticipation. A complexity theory of design may also help us understand what is complex and where it is needed. It is also worth noting that there is a tradition in design disciplines (which might not be recorded in scientific papers) to use complexity concepts in designing practice and discourse more loosely – as an inspiration and as source for creativity and innovation.

5. LINKING DESIGN AND COMPLEXITY – NEW DIRECTIONS

It can be argued that all the above approaches to linking design with complexity are mainly applications of complexity concepts and measurements in design research and practice. One interesting research theme would be to focus on the opposite direction of exchange and investigate what design can offer to complexity research. The proposal brought forward is that design is a class of research problem fundamentally linked with the complexity of a system and we therefore need to devise formal theories of design as an organizational capacity[122].

To consider design as a natural capacity of organization is a novel but potentially controversial statement for two reasons.

First, the view implies that design is intelligible and explainable by natural processes, independently from the explicit recognition of an intelligent agent. This is contrast to most design research which is dominated by the cognitive stance: empirical studies on design thinking, computational tools or theoretical methods for design problem solving and models/simulations of design processes typically assume the authority of a cognitive design agent. However, the statement is not so contentious from a methodological point of view. Take for instance the human brain which is the best example of a design-capable system. The ability to design can be explained by the complexity and organization of the brain independently from the overall -intelligent- function of the system. The controversy however arises as the assertion also implies an epistemological generalization: If design is a capacity derived from the organization of a system then other realizations of complex systems are possible embodiments of design-capable systems. In short, design is not simply the science of the artificial.

Second, complexity science as a science of effective organization normally assumes design as a top-down, externally imposed explanation. It is not common to consider that design is a natural characteristic explained by the organization of a system. As we saw, complexity research is concerned with the development of theories about the origins and characteristic capacities of systems (such as life and cognition), where design is assumed as an alternative explanatory stance. Elaborating a ("complexity") theory of design capacity of systems is therefore a two-sided contribution: it is a methodological contribution to the study of design but also a potentially interesting problem within the complexity research agenda.

REFERENCES

COMPLEXITY RESEARCH

1. C.E. Shannon, "A mathematical theory of Communication," *The Bell System Technical Journal.* **27**, pp. 379-423, 623-656,1948.
2. Y. Bar-Yam, *Dynamics of Complex Systems,* Addison-Wesley, New York, 1997.
3. A.-L. Barabási, R. Albert, "Emergence of scaling in random networks", *Science* **286**, 509-512, 1999.
4. M. Batty, *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-based Models and Fractals*, MIT Press, Cambridge Mass, 2005.
5. M. Faloutsos, P. Faloutsos, C. Faloutsos, C, "On power law relationships of the internet topology", *Computer Communication Review.* **29**, 251-262, 1999
6. K-I Goh, E.S. Oh, H. Jeong, B Kahng, D. Kim, "Classification of scale free networks", *Proceedings of the National academy of Science of US,* **99**, pp. 12583-12588, 2002.
7. G Troll, P beim Graben, "Zipf's Law is not a consequence of the central limit theorem", *Physical Review E.* **57**, pp. 1347-1355, 1998.
8. G.K. Zipf, *Human behavior and the principle of least effort,* Addison-Wesley, New York 1949.
9. B. B Mandelbrot, *The Fractal Geometry of Nature*, Freeman, New York, 1983.
10. R.H. Atkin, *Mathematical structure in Human Affairs*, Heinemann Educational Books, London 1974.
11. J. Johnson, "The multidimensional networks of complex systems", in *Networks in Action: Communication, economics and human Knowledge,* D. Batten, J. Casti, R. Thord eds., Springer-Verlag, 1995.

12. E. Estrada, J.A. Rodriguez-Velazquez, *Complex networks as hypergraphs.* [Available online at http: //arxiv.org/ftp/physics/papers/0505/0505137.pdf]
13. A.C. Ehresmann, J-P. Vanbremeersch, "Emergence processes up to consciousness using the multiplicity principle and quantum physics", in *Computing Anticipatory Systems (CASYS 2001) Conference Proceedings*, American Institute of Physics, 2002.
14. G. J. Chaitin, “On the Length of Programs for Computing Finite Binary Sequences”, *Journal of the Association of Computing Machinery*, **13**, pp. 547-569, 1996.
15. G.J. Chaitin, *The limits of mathematics: A course on information theory and limits of formal reasoning,* Springer-Verlag, Singapore, 1997.
16. G.W. Flake, *The computational beauty of Nature: Computer explorations of fractals, chaos, complex systems and Adaptation,* MIT Press, Cambridge, 1998.
17. A. Kolmogorov, “Three Approaches to the quantitive definition of Information”, *Problems of Information Transmission*, **1**, pp. 1-17, 1965.
18. M. Li, P Vitányi, *An Introduction to Kolmogorv Complexity and Its Applications*, Springer, 1997
19. C.H. Bennett, “Dissipation, Information, computational complexity and the definition of organization ”, in *Emerging Syntheses in Science*, D. Pines ed., pp. 215-233, Addison-Wesley, 1985.
20. C.H. Bennett, “How to Define Complexity in Physics, and Why”, in C*omplexity, Entropy and the Physics of Information* W H Zurek, ed., pp. 137-147, Addison-Wesley, 1990.
21. B. Edmonds, “What is Complexity?-The philosophy of complexity per se with application to some examples in evolution”, *in The Evolution of Complexity,* F. Heylighen and D. Aerts eds., Kluwer, Dordrecht, 1999.
22. N. Boccara, *Modeling Complex Systems*, Springer-Verlag, New York, 2004.
23. J. P. Crutchfield, J. D. Farmer, N. H. Packard, and R. S. Shaw, “Chaos”, *Scientific American*, **255**, pp. 46-57, 1986.
24. S. Kauffman, *At Home in the Universe: a search for the laws of self-organization and complexity*, Oxford University Press, New York, 1995.
25. P. Bak, C. Tang, K. Wiesenfeld, “Self-organized criticality”, *Physical Review A,* **38**, 364-374, 1988.
26. W. B. Arthur, “Competing Technologies, Increasing Returns, and Lock-In by Historical Events”, *Economic Journal,* **99**, pp. 116-131, 1989.
27. S. J. Liebowitz and S. E. Margolis, “Path Dependence, Lock-in, and History”, *Journal of Law, Economics, and Organization*, **10**, pp. 205-226, 1995.
28. T.C. Schelling, “Dynamic Models of Segregation,” *Journal of Mathematical Sociology,* **1**, pp. 143-186, 1971.
29. S. Wolfram, *Cellular Automata and Complexity,* Addison-Wesley, Reading, Mass, 1994
30. S. J, Gould, *The structure of Evolutionary Theory*, Belknap Press of Harvard University Press, Cambridge 2002.
31. R. M. Axelrod, and W. D. Hamilton, “The Evolution of Cooperation”, *Science*, **211**(4489), pp. 1390-1396, 1981.
32. R. M. Axelrod, The Complexity of Cooperation: agent-based models of competition and collaboration, Princeton University Press, Princeton, 1997.
33. G. Nicolis, and I. Prigogine, *Self-Organization in Non-Equilibrium Systems*, Wiley, New York, 1977.
34. P. Bak, *How Nature Works: the science of self-organized criticality,* Copernicus , New York, 1996.
35. H. J. Jensen, *Self-organized Criticality: emergent complex behaviour in physical and biological systems,* Cambridge University Press, Cambridge, 1998.
36. H. Haken, *Synergetics: An introduction. Nonequilibrium phase transitions and self-organization in physics, chemistry and biology*, Springer Verlag, Berlin, 1977.
37. T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, 1984.
38. D. E. Rumelhart and J. L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, (Volume 1: Foundations), MIT Press, Cambridge, MA, 1986.
39. R. F. Port and T. van Gelder. *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge, MA, 1995.
40. R. D. Beer, “A dynamical systems perspective on agent-environment interaction”, *Artificial Intelligence,* **72**, pp. 173-215, 1995.
41. J. Collier, “What is autonomy?”, *International Journal of Computing Anticipatory Systems* (CASYS 2001), 2002. [last accessed online, June 2002, at http://cogprints.ecs.soton.ac.uk/archive/00002289/]
42. W. D. Christensen, and C. A. Hooker “Anticipation in autonomous systems: foundations for a theory of embodied agents”, *International Journal of Computing Anticipatory Systems*, **5**, pp. 135-154, 2000.
43. T. Smithers, “Autonomy in robots and other agents”, *Brain and Cognition*, **34**, pp. 88-106, 1997.
44. P. Cariani, *On the Design of Devices with Emergent Semantic Function*, PhD Thesis, New York State University, 1989.

45. A. Cangelosi, and D. Parisi, “The emergence of a ‘language’ in an evolving population of neural networks”, *Connection Science*, **10**, pp. 83-97, 1998.
46. B. MacWhinney, “Models of the emergence of language”, *Annual Review of Psychology*, **49**, pp. 199-227, 1998.
47. L. Steels, “Evolving grounded communication for robots”, *Trends in Cognitive Sciences*, **7**, pp. 308-312, 2003.
48. S. Kauffman, 'Metabolic stability and epigenesis in randomly connected nets', *Journal of Theoretical Biology*, **22**, 437-467, 1969.
49. H.R. Maturana, F.J Varela, *Autopoiesis and Cognition*: *The Realization of the Living,* D. Reidel Publishing, 1980.
50. F. Varela, *Principles of Biological Autonomy*, Elsevier/North-Holland, New York, 1979.
51. R. Rosen, *Life itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life,* Columbia University Press, 1991.
52. N. H. Gregersen (ed.), *From Complexity to Life: On the Emergence of Life and Meaning,* Oxford University Press, Oxford, 2003.
53. N. Luhmann, *Essays on Self-reference*, Columbia University Press, New York, 1990.
54. C. Emmeche, *The Garden in the Machine. The Emerging Science of Artificial Life*, Princeton University Press, Princeton, 1994.
55. C. Langton, “Artificial Life”, in *Artificial Life*, C. Langton, ed., SFI Studies in the Sciences of Complexity, Proc. Vol. VI. Addison-Wesley, Redwood City, CA, 1989.
56. F. J. Varela, and P. Bourgine, eds., *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, The MIT Press, Cambridge, MA, 1992.
57. L. Steels, and R. Brooks eds., *The `artificial life' route to `artificial intelligence'. Building Situated Embodied Agents*. Lawrence Erlbaum Associates, New Haven, 1994.

DESIGN RESEARCH

58. H. A. Simon, *The Sciences of the Artificial*, The MIT Press, Cambridge MA, 1996.
59. P. G. Rowe, *Design Thinking*, The MIT Press, Cambridge Mass, 1987
60. B. L. Archer, “An overview of the structure of the design process”, in *Emerging Methods in Environmental Design and Planning*, G.T. Moore ed., pp. 285-307, MIT Press, Cambridge Mass., 1970.
61. M. Asimov, *Introduction to Design*, Prentice Hall, New Jersey, 1962.
62. N. Cross ed. *Developments in Design Methodology*, John Wiley, Chichester, 1984.
63. J. C. Jones and D. Thornley ed., *Conference in Design Methods*, Pergamon, Oxford, 1963.
64. G. T. Moore, *Emerging Methods in Environmental Design and Planning*, MIT Press, Cambridge Mass., 1970.
65. A. Newell, J. C. Shaw, and H. A. Simon, *Elements of a Theory of Problem Solving*, Rand Corporation Report P-971, 1957.
66. Ö. Akin, *Psychology of Architectural Design*, Pion Ltd, London, 1986.
67. C. M. Eastman, “On the analysis of the intuitive design process”, in *Emerging Methods in Environmental Design and Planning*, G. T. Moore ed., pp. 21-37, MIT Press, Cambridge Mass., 1970.
68. C. M. Eastman, “Automated space planning”, *Artificial Intelligence*, **4**, pp. 41-64, 1973.
69. U. Flemming, C. A. Baykan, R. F. Coyne, R.F., “Hierarchical generate-and-test versus constraint-directed search”, in *Proceedings of the Artificial Intelligence in Design*, J. Gero ed., Kluwer, Dordrecht, 1992.
70. A. H. B. Duffy, “The ‘what’ and ‘how’ of learning in design”, *IEEE Expert*, **12**, pp. 71-76, 1997.
71. J. S. Gero, “Design tools that learn”, *Advances in Engineering Software*, 29, pp. 755-761, 1998.
72. D. L. Grecu, and D. C. Brown, “Dimensions of machine learning in design”, *AIEDAM*, **12**, pp. 117-121, 1998.
73. Y. Reich, “Learning in design: from characterizing dimensions to working systems”, AIEDAM, 12, pp. 161-172, 1998.
74. L. Candy, “Representations of strategic knowledge in design”, *Knowledge Based Systems*, **11**, pp. 379-390, 1998.
75. G. Carrara and Y. Kalay, *Knowledge-Based Computer-Aided Architectural Design*, Elsevier, Amsterdam, 1994.
76. D. C. Brown and B. Chandrasekaran, *Design Problem Solving: Knowledge structures and control strategies*, Morgan Kauffman, San Mateo CA, 1989.
77. M. L. Maher, J. Poon, S. Boulanger, “Formalizing design exploration as co-evolution”, in *Advances in Formal Design Methods for CAD*. Proceedings of the IFIP WG5.2 workshop on formal design methods for Computer-Aided Design, Gero, J.S., Sudweeds, F. eds., Chapman and Hall, London, 1996.
78. M. L. Maher, “A model of co-evolutionary design”, *Engineering with Computers*, **16**, pp. 195-208, 2000.
79. R. D. Coyne and S. Newton, “Design reasoning by association”, *Environment and Planning B*, **17**, pp. 39-56, 1989.
80. R. D. Coyne, “Design reasoning without explanation”, *AI Magazine*, **11**, pp. 72-80, 1990.

81. J. S. Gero, "Recent design science research: constructive memory in design thinking", *Architectural Science Review*, **42**, pp. 3-5, 1999.
82. A. Goel, *Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving*, PhD Thesis, Department of Computer and Information Science, The Ohio State University, 1989.
83. E. A. Domeshek and J. L. Kolodner, "Case-based design aid for architecture", in *Artificial Intelligence in Design '92*, J. S. Gero and F. Sudweeks eds., Kluwer Academic Publishers, Dordercht, 1992.
84. U. Flemming, "Case-based design in the SEED system", in *Knowledge-Based Computer-Aided Architectural Design,* G. Carrara and Y. Kalay eds., Elsevier, Amsterdam, 1994.
85. M. L. Maher and P. Pu eds., *Issues and Applications of Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, Mahwah, NJ, 1997.
86. R. E. Oxman, "Precedents in Design: a Computational Model for the Organization of Precedent Knowledge", *Design Studies*, **15**, 141-157, 1994.
87. F.M.T Brazier, L. V. Moshkina, N. J. E. Wijngaards, "Knowledge level model of an individual designer as an agent in collaborative distributed design", *Artificial Intelligence in Engineering*, **15**, pp. 137-152, 2001.
88. T. Smithers, "On knowledge-level theories of design process", in *Artificial Intelligence in Design '96*, J. S. Gero and F. Sudweeks eds., Dordrecht, Amsterdam, 1996.
89. T. Smithers, "Towards a knowledge-level theory of design process", in *Artificial Intelligence in Design '98,* Gero and F. Sudweeks eds., Dordrecht, Amsterdam, 1998.
90. C. Alexander, *Notes on the Synthesis of Form*, Harvard University Press, Cambridge Mass, 1964.
91. M. Batty and P. A. Longley, *Fractal Cities: a Geometry of Form and Function*, Academic Press, San Diego CA, 1994.
92. M. Batty and Y. Xie, "Possible urban automata", *Environment and Planning B*, **21**, pp. 175-192, 1997.
93. B. Hillier, Space is the Machine, Cambridge University Press, Cambridge, 1996.
94. J. Cagan and W. J. Mitchell, "Optimally directed shape generation by shape annealing", *Environment and Planning B*, **20**, pp. 5-12, 1993.
95. S. Chase, "Logic Based Design Modeling With Shape Algebras", *Automation in Construction,* **6**, pp. 311-322, 1997.
96. C. Earl, "Creating design worlds", Environment and Planning B: Planning and Design, 13, pp. 177-188, 1986.
97. C. Earl and J. Johnson, "Graph theory and Q-analysis", *Environment and Planning B*, **8**, pp. 367- 391, 1981.
98. U. Flemming and S. F. Chien, "Schematic layout design in SEED environment", *Journal of Architectural Engineering*, **1**, pp. 162-169, 1995.
99. T. Knight, "Shape grammars and color grammars", *Environment and Planning B*, **21**, pp. 705-735, 1994.
100. R. Krishnamurti, "The arithmetic of shapes", *Environment and Planning B: Planning and Design*, **7**, pp. 463-484, 1980.
101. L. March "A Boolean description of a class of built forms", in *The Architecture of Form*, March L. ed., Cambridge University Press, Cambridge, 1976.
102. W. J. Mitchell, P. Steadman, and R.S. Liggett, "Synthesis and optimization of small rectangular floor plans", Environment and Planning B, 3, pp. 37-70, 1976.
103. P. Steadman, *The Automatic Generation of Minimum-Standard House Plans*, Centre for Land Use and Built Form Studies, Working Paper, University of London, 1970.
104. G. Stiny, "Introduction to shape and shape grammars", *Environment and Planning B*, **7**, pp. 343-351, 1980.
105. G. Stiny, "The algebras of design", *Research in Engineering Design*, **2**, pp. 171-181, 1991.
106. D. A. Schön, *The Reflective Practitioner. How professionals think in action*, Basic Book, New York, 1983.
107. L. L. Bucciarelli, *Designing Engineers*, MIT Press, London, 1994.
108. N. E. Branki, E. A. Edmonds, R. M. Jones, "A study of socially shared cognition in design", *Environment and Planning B: Planning and Design,* **20**, pp. 295-306, 1993.
109. N. Cross, and A. C. Cross, "Observations of teamwork and social processes in design", *Design Studies*, **16**, pp. 143-170, 1995.
110. M. Cumming, "A bottom up approach to design coordination", in *Proceedings of the 6th International Conference in Design and Decision Support Systems in Architecture*, Eindhoven University of Technology, 2002.
111. G. Fischer, A. C. Lemke, R. McCall, A. I. Morch, "Making argumentation serve design", in *Design Rationale: Concepts, Techniques, and Use*, T. P. Moran, Carroll, J.M. eds., pp. 267-293, Lawrence Erlbaum Associates, Mahwah, NJ, 1996.
112. M. D. Gross, E.Y-L Do, R. J. McCall, W. V. Citrin, P. Hamill, A. Warmack, K. S. Kuczun,. "Collaboration and coordination in architectural design: Approaches to computer mediated team work", *Automation in Construction*, **7**, pp. 465-473, 1998.
113. T. Kvan, (1999). *Designing Together Apart. Computer supported collaborative design in architecture*, PhD Thesis, The Open University, 1999.

114. R. Sosa and J. S. Gero, "A computational study of creativity in design: the role of society", *AIEDAM,* 2005 [to appear]
115. S. C. Stumpf, J. T. McDonnell, "Talking about team framing: using argumentation to analyse and support experiential learning in early design episodes", *Design Studies,* **23**, pp. 5-23, 2001.
116. B. Trousse, H. Christiaans, "Design as a topos-based argumentative activity: a protocol analysis study". *Analysing Design Activity*. N. Cross, Christiaans, H., Dorst, K. Chichester, eds., Wiley, 1996.
117. M. Perry, D. Sanderson, "Coordinating joint design work: the role of communication and artifacts", *Design Studies,* **19**, pp. 273-288, 1998.
118. G. Fischer, "Social creativity, symmetry of ignorance and Meta-Design", in *Proceedings of the Third Conference on Creativity and Cognition*, Loughborough, UK, ACM, 1999.
119. E. Arias, H. Eden, G. Fischer, A. Gorman, E. Scharff, "Transcending the individual human mind - creating shared understanding through collaborative design", *ACM Transactions on Computer-Human Interaction,* **7**, pp. 84-113, 2000.
120. J. Hollan, E. Hutchins, D. Kirsh,"Distributed cognition: toward a new foundation for human-computer interaction research", *ACM Transactions on Computer-Human Interaction,* **7**, pp. 174-196, 2000.
121. T. Zamenopoulos, K. Alexiou, "Computer-aided creativity and learning in distributed cooperative human-machine networks", in *CAAD Futures 2003*, Kluwer Academic Publishers, Dordrecht, 2003.
122. T. Zamenopoulos and K. Alexiou, "The problem of design in complexity research", in *Proceedings of the ECCS Conference*, 2005 [to appear].
123. M. Batty, "Limits to prediction in science and design science", *Design Studies*, **1**, pp. 153-159, 1980.
124. R. Coyne, A. Snodgrass, "Is designing mysterious? Challenging the dual knowledge thesis", *Design Studies,* **12**, pp.124-131, 1991.
125. N. Cross, "Designerly ways of knowing", *Design Studies*, **3**, pp. 221-227, 1982.
126. K. Friedman, "Theory construction in design research: criteria: approaches, and methods", *Design Studies,* **24**, pp. 507-522, 2003.
127. B. Lawson, *How Designers Think: the design process demystified*, Architectural Press, Oxford, 1997.
128. T. Love, "Philosophy of design: a meta-theoretical structure for design theory", *Design Studies*, **21**, pp. 293-313, 2000.
129. P. Steadman, "The history and science of the artificial", *Design Studies*, **1**, pp. 49-58, 1979.

DESIGN AND COMPEXITY

130. S. Austin, A. Newton, J. Steele and P. Waskett, "Modelling and managing project complexity", *International Journal of Project Management*, **20**, pp. 191-198, 2002.
131. A. Calinescu, J. Efstathiou, J. Schirn, J. Bermejo, "Applying and assessing two methods for measuring complexity in manufacturing", *The Journal of the Operational Research Society*, 49, pp. 723-733, 1998.
132. C. Earl, J. Johnson and C. Eckert, "Complexity", in *Design Process Improvement. A review of current practice*, J. Clarkson and C. Eckert eds., pp. 174-197, Springer-Verlag, London, 2005.
133. C. Eckert, P. J. Clarkson, W. Zanker, "Change and customization in complex engineering domains", *Research in Engineering Design*, **15**, pp. 1-21, 2004.
134. N. P. Suh, "A theory of complexity, periodicity and the design axioms", *Research in Engineering Design*, **11**, pp. 116-131, 1999.
135. J. Gero and V. Kazakov, "On measuring the visual complexity of 3D solid objects", *Journal of Design Sciences and Technology*, 2005 [to appear].
136. A. Koutamanis, "Modeling irregular and complex forms", In *AVOCAAD - Added Value of Computer Aided Architectural Design*, N. Koenraad, P. Tom, V. Johan, V. Johan eds., 2001.
137. O. Maimon and D. Braha, "On the complexity of the design synthesis problem", *IEEE Trans. on Systems, Man and Cybernetics*, 26, pp. 142-151. 1996.
138. M. Klein, H. Sayama, P. Faratin and Y. Bar-Yam, "A complex systems perspective on computer-supported collaborative design technology", *Communications of the ACM*, 45, pp. 27-31, 2002.
139. J. H. Johnson, "The multidimensional networks of complex systems", in *Networks in Action*, D. Batten, J. Casti, R. Thord eds., pp. 49-79, Springer-Verlag, Berlin, 1995.
140. J. H. Johnson, "The multilevel multidimensional networks of complex urban systems", in *CUPUM 05*, S. Batty ed., CASA, 2005.
141. M. Batty, *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-based Models and Fractals*, MIT Press, Cambridge Mass, 2005.
142. S. Marshall, *Streets and Patterns*, Spon Press, London 2004.
143. P. J. Bentley ed., *Evolutionary Design by Computers*, Morgan Kaufmann Publishers Inc, San Francisco, CA, 1999.
144. J. Frazer, *An Evolutionary Architecture*, Architectural Association, London, 1995

145. I. Parmee, “Adaptive Computing in Design and Manufacture”, *IEEE Trans. Evolutionary Computation,* **3**, pp. 159-159, 1999.
146. H. Achten, “An approach for agent-systems in design support”, in Intelligent Agents in Design Workshop notes, AID '02 Conference, Cambridge, UK, 2002.
147. M. I. Campbell, J. Cagan, K. Kotovsky, K. “A-Design: Theory and implementation of an adaptive, agent-based method for conceptual design”, *Artificial Intelligence in Design '98*, J. S. Gero, Sudweeks, F. eds., pp. 579-598, Kluwer Academic Publishers, Dordrecht, 1998.
148. P. Coates, R. Thum, “Parallel systems and architectural form”, in *Proceedings of Greenwich 2000: Digital Creativity Symposium,* Greenwich, University of Greenwich, 2000.
149. E. A. Edmonds, L. Candy, R. Jones, B. Soufi, “Support for collaborative design: agents and emergence” *Communications of the ACM,* **37**, pp. 41-47, 1994.
150. J. S. Gero, and R. Sosa, R., “Creative design situations: Artificial creativity in communities of design agents”, *CAADRIA 2002*, Prentice Hall, New York, 2002.
151. P. Leclercq, and R. Juchmes, “The absent interface in design engineering”, *AIEDAM*, Special Issue: Human-Computer Interaction in Engineering Contexts, I. C. Parmee, I. F. C. Smith, eds. **16**, 2002.
152. G. Rzevski, “A framework for designing intelligent manufacturing systems”, *Computers in Industry*, **34**, pp. 211-219, 1997.
153. D. Saarloos, T. Arenze, A. Borges, H. Timmermans, “A multi-agent model for alternative plan generation”, *Environment and Planning B: Planning and Desig*n, **32**, pp. 505-522, 2004.
154. R. Saunders and J. S. Gero “Curious agents and situated design evaluations” *Agents in Design 2002*, J. S. Gero, F. M. T. Brazier, eds., pp. 133-149, Key Centre for Design Computing and Cognition, University of Sydney, Sydney, 2002.
155. T. Zamenopoulos, and K. Alexiou, “Structuring the plan design process as a coordination problem: the paradigm of distributed learning control coordination”, in *Advanced Spatial Analysis: the CASA Book of GIS*, P. Longley and M. Batty eds., pp 407-426, ESRI Press: Redlands, 2003.
156. E. Besussi, and A. Cecchini, eds., *Artificial Worlds and Urban Studies*, DAEST, IUAV, Venice, 1996.
157. R. White,and G. Engelen, “High-resolution integrated modelling of the spatial dynamics of urban and regional systems” *Computers, Environment and Urban Systems,* **24**, pp. 383-400, 2000.
158. J. Portugali and H. Casakin, “SIRN (Synergetic Inter-Representation Network): an approach to design”, in *Common Ground: Design Research Society International Conference*, Durling, D., Shackleton, J. (eds), Staffordshire University Press, Staffordshire, 2002.
159. T. Zamenopoulos, and K. Alexiou, “Design and Anticipation: towards an organisational view of design systems’. CASA Working Paper 76, University College London, Centre for Advanced Spatial Analysis, 2004. [Available from URL: http://www.casa.ucl.ac.uk/working_papers/paper72.pdf]

ISBN: 978-0-74921-545-3

Published by the Open University