Design as conversation with digital materials

DEARDEN, Andy <http://orcid.org/0000-0002-5706-5978>

Cover sheet

Designing as a conversation with digital materials

Andy Dearden

Communications & Computing Research Centre
Sheffield Hallam University
Howard St., Sheffield
S1 1WB

a.m.dearden@shu.ac.uk

Tel: +44 114 225 2916
Fax: +44 114 225 3161

Keywords:

Computer supported design, Design tools, Psychology of design, Philosophy of design, Design theory.

Abstract

This paper explores Donald Schön's concept of design as a conversation with materials, in the context of designing digital systems. It proposes *material utterance* as a central event in designing. A material utterance is a situated communication act that depends on the particularities of speaker, audience, material and genre.

The paper argues that, if digital designing differs from other forms of designing, then accounts for such differences must be sought by understanding the material properties of digital systems and the genres of practice that surround their use. Perspectives from human-computer interaction (HCI) and the psychology of programming are used to examine how such an understanding might be constructed.

Donald Schön (Schön, 1995; Schön & Bennet, 1996) suggests a characterisation of designing as a *conversation with materials*. Digital systems are increasingly common, both as the outcomes of designing and as artefacts used within designing activity. It is therefore useful, in developing accounts of digital designing, to consider the nature of conversations that include the manipulation of digital materials.

In this paper, I draw upon linguistic accounts of the nature of conversation and dialogue. Using concepts from linguistic pragmatics and from literary theory, in particular notions of 'utterance' and 'speech genre', I develop an account of designing in relation to 'material utterance'.

I then use this account to review relevant research in human-computer interaction (HCI), and the psychology of programming. Specifically, I consider research that examines: the experience of interacting with notational systems and aspects of interacting with formal systems. I suggest that these areas of research provide a useful resource from which more detailed accounts of digital designing in particular domains might be constructed.

1 Pragmatic accounts of conversation

The use of linguistic pragmatics to support the development of theories of designing is well established (Lloyd, 2000; Bucciarelli, 2002; Coyne et al., 2002). Linguistic pragmatics takes as its subject of study, the actual use, co-ordination and interpretation of language in practical interaction between people (Levinson 1983). From a pragmatic perspective, the key elements of language are not words and sentences. Instead, pragmatics introduces techniques such as 'conversation analysis' (Hutchby & Wooffitt, 1998), the study of 'speech acts' (Searle, 1969), or analysis of 'discourse' (Yates et al., 2001). A key concept in pragmatics is the *utterance* (Austin, 1962; Fish, 1980). To contrast with a sentence, an utterance is a specific instance of a specific speaker addressing specific audience (that may be immediately present or otherwise). An utterance is always situated in a particular context. An utterance always includes some expectation of the response from the audience. Such discussions draw attention to the

way in which the form and meaning of our utterances depends upon the context in which the utterance is made.

In considering how the experience and history of speaker and audience impact on the form of utterances, the Russian literary theorist Mikhail Bakhtin (and colleagues) propose the concept of *speech genre* (Morris, 1994; Bakhtin, 1994). Bakhtin contrasts the example of military commands with their archetypal intonation and tightly defined expectation of audience response (obedience), from the very different utterances, open to much broader interpretation, shared between intimates. Speech genres are learned socially through interaction. Bakhtin's speech genres can be related to Wittgenstein's *language games* (Kerry, 1994). Speech genres (and language games) are more than merely styles of speaking. Different speech genres lend themselves to expression of different facets of our experience; they are associated with particular ways of seeing the world, highlighting certain aspects whilst ignoring others (Morson & Emerson, 1990). Bakhtin argues that the meaning of an utterance can never be *finalised*, that is, there is no final oracle to which we can appeal to identify a correct, incontestable interpretation of the utterance. Instead, every utterance implicates a *field of answerability,* a social and historical space in which the factors that shaped the utterance and the consequences of the utterance can always be (recursively) questioned. Hence, Bakhtin implicates the audience (as well as all the utterances that the speaker has previously made or heard) in both the form and the meaning of the utterance. The works of Bakhtin & colleagues have recently attracted the attention of researchers in human-computer interaction (Spinuzzi, 1999; McCarthy & Wright, 2004; Wright & McCarthy, 2004), as well as from digital artists (1999) and commentators in the humanities (Rockwell, 2002).

These discussions of utterance, thus provide a framework that emphasises the way that social, cultural and contextual factors influence both the form and the meaning of an utterance. Liddament (1999) highlights the importance of considering such issues in developing adequate theories of designing. Figure 1 illustrates the factors that shape the form and meaning of an utterance in this account. Note that the utterance is both shaped by genre, speaker and audience, and impacts on the development of speaker, audience and genre.

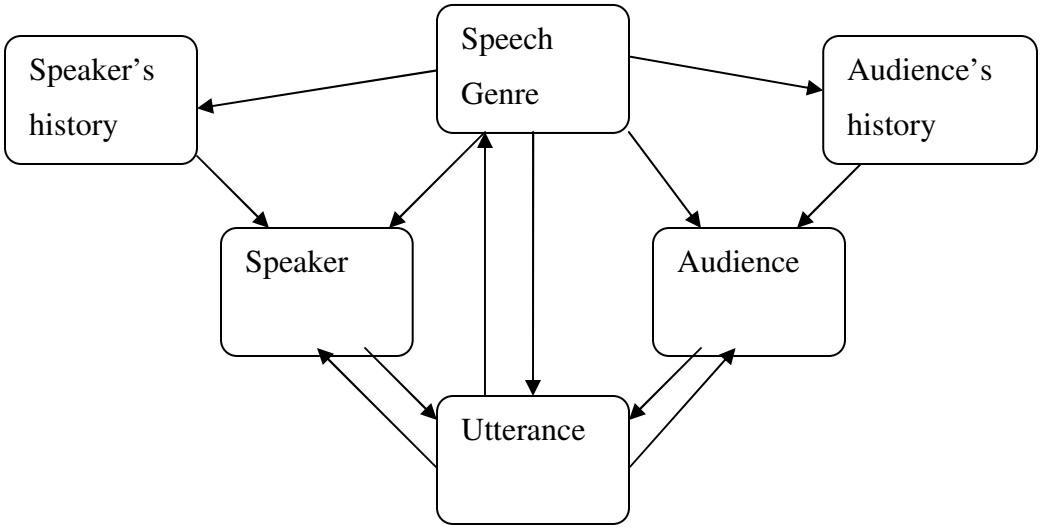

Figure 1: factors that shape the form of an utterance

2 Designing and material utterance

Drawing on the concept of utterance, together with Schön's metaphor of a conversation with materials, I propose the concept of *material utterance,* that is an utterance that includes the making or modification of material artefacts (including representational artefacts such as files in a digital system). Figure 2, extends the diagram in figure 1 by adding a material element to the utterance. As with any utterance, the form and meaning of the utterance are shaped by the speaker, audience, history and genre, but additionally by the materials available in making the utterance. This view has parallels with the observations of Lloyd & Snelders (2003) on the creative role of both designer and audience in successful designing. Note that the material utterance also alters that stock of materials available in forming future utterances.

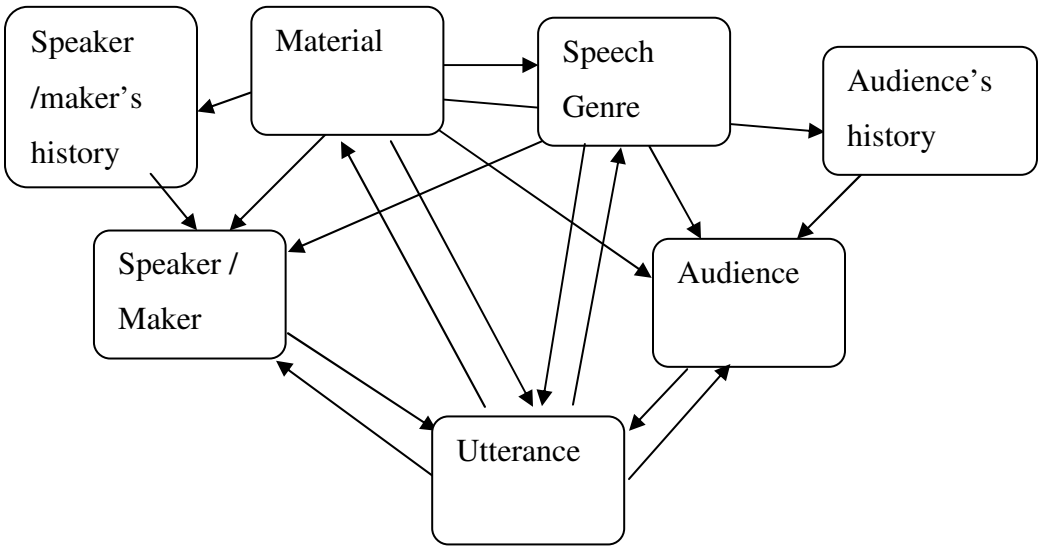

Figure 2: factors that shape the form of a material utterance

I then propose a working definition of designing activity as:

> the making of utterances (material or otherwise) that are oriented towards the form of future utterances to involve a different audience.

In this view of designing, a customer giving a design brief to a designer, a designer showing a prototype to a customer, a requirements engineer presenting a specification document to clients, an industrial designer setting up a computer aided cutting tool, or a jeweller presenting sketches in response to a commission, are all material utterances and are all designing activities. This definition is intended to incorporate the idea of future intent that is generally accepted as part of designing. On the other hand, the definition does not exclude utterances that do not include a persistent material element. Hence, talking about the form of a prototype is a designing activity by this definition, and rehearsing a play is also a designing activity. Similarly, a user or client making comments about a prototype, or suggesting modifications to a prototype would also be categorised as a designing activity.

Viewing design in terms of material utterance and genre offers the interesting possibility of using a uniform conceptual framework for considering both: utterances that are made in the activity of designing, for example utterances made by designers and users or by

specialists with different disciplines who are contributing to designing; and utterances that are the outcomes of designing activity, for example the release of a new product or technology. The latter perspective reflects observations from authors such as Carroll *et al*. (1991) of the 'task-artifact cycle' in which new computing artefacts give rise to desires or needs for new tasks, and Woolgar's (1994) perspective of 'the system as text'. In the rest of this paper, I shall focus primarily on the first category, i.e. utterances made within designing activity. Where I intend to consider utterances and genres relating to the outcomes of designing, I shall explicitly indicate this.

A critical aspect in the distinction of material utterances from the general case, is that all materials have properties that moderate the ease or difficulty of creating a material utterance. For example: different materials have different density, ductility, plasticity and malleability; wood has a grain and different species of wood have different strengths and weaknesses; different paint types lend themselves to different visual effects; manufacturing processes enable some forms to be more readily produced than others; existing parts may be more or less readily available. Doorst & Cross (2001) describe how designers' explorations of the potential solution space in response to a design brief result in reconceptualisation and reconfiguration of the identified 'design problem' to be solved. Schön suggests a metaphor in which materials 'talk-back' to the designer. A similar observation has been made in relation to large-scale software development. Bowers & Pycock (1994) discuss 'gradients of resistance' in designing, noting how clients or users will offer contributions to a design discussion not as 'requirements' that they have, but as queries as to what might be possible. By this means, the 'requirements' of a system are negotiated "… as a product of argument and *resistance*" (my italics). Understanding such properties, either in terms of explicit 'facts' or as tacit knowledge gained through experience, impacts upon a person's skill in constructing material utterances. Kapor (1991) points to the need for software designers to develop a critical understanding of the properties of digital media. Hence, in digital designing, as much as in any other area of designing, it becomes important to understand the material properties of the (digital) medium. To re-iterate, at this level of abstraction, there is no distinction between digital designing and other forms of designing. The challenge of developing theories of digital designing or digital design thinking is to understand the material properties of the digital

media involved, and the consequences for material utterances that result from those properties.

A further point to recognise is that, since a material utterance is a special case of an utterance, it is also the case that the audience (perhaps users or clients) and the speech genre are involved in the form and meaning of the utterance. Understanding of audiences and genres will also impact upon a person's skill in making material utterances. This means that designing must be understood as involving both conversations with materials, conversations with an audience using materials, and conversations conducted in the context of genre.

Lastly, the interpretation of a material utterance can never be finalised. Although a genre may suggest a particular interpretation and required response (disallowing alternative responses) there is always the possibility of *surprise* and the mechanisms to establish and maintain the fixed interpretation are ultimately socio-political. For example, a contract formally specifying particular requirements and deadlines for some to be designed artefact may evoke the response of producing an artefact meeting the requirements or might provoke a legal dispute or even more radical responses. Similarly, a designed artefact may give rise to surprising responses from its audience (users). For example, consider the way that text messaging in mobile phones, initially expected to be a minor function of interest to a very limited number of specialist users, has in fact led to the creation of multiple rich genres of social communication. Bakhtin suggests that when different genres come into contact, this offers a particularly rich environment for creative understanding and surprise (see Wright & McCarthy 2005 for a discussion). Doorst & Cross (2001) & Schön (1995) both suggest an important relationship between surprise and creative designing.

In the rest of this paper, I examine research in human-computer interaction and the related area of the psychology of programming. These perspectives provide some useful insights into the material characteristics of digital media and the potential impact of these characteristics on the making and interpretation of utterances in digital designing.

3 Digital media as notational systems

Many digital systems used to support designing activity can be classed as notational systems, that is, systems that allow utterances to be formed using a restricted set of symbols, and constrained syntax, that have (socially) agreed interpretations. Computer programming languages can be considered as an extreme example of such notational systems, where the level of agreement regarding the interpretation is strongly moderated and replicated by the production of software such as programming environments and compilers.

3.1 Cognitive Dimensions and conversations with materials

In the psychology of computer programming, Green & colleagues (Green & Petre, 1996; Green & Blackwell, 1998; Blackwell & Green, 1999; Blackwell & Green, 2000) have developed the 'cognitive dimensions' framework for assessing features that impact on the usability of notational systems. The framework includes a vocabulary that enables discussion of the properties of notational systems, by providing "… a small number of terms that have been chosen to be easy for non-specialists to comprehend, whilst capturing a significant amount of the psychology and HCI of programming." (Green & Petre, 1996, p138). The framework allows discussion both of notations as independently defined systems (e.g. the Java programming language) and tools that are available to manipulate such notations, e.g. integrated software development environments. As well as the set of terms for the dimensions, the framework offers additional conceptual tools, such as: a concept of 'layers' that considers the relation between a notation, and specific tools for editing that notation; and a concept of 'helper–devices' that may be included in an editing environment to support specific transformations (e.g. a search and replace function in a word processor can be treated as a helper device). A complete discussion of the framework is beyond the scope of this paper. However, some examples of the dimensions serve to illustrate the approach.

The first example dimension, that reflects an experience that will be familiar to any regular user of digital design tools, is *Viscosity*, which Green & Blackwell (1998) define as "resistance to change, the cost of making small changes." Green & Blackwell draw attention to two different types of viscosity, namely: repetition viscosity, in which "a

single goal-related operation on the information structure (one change 'in the head') requires an undue number of individual actions"; and knock-on viscosity in which "one change 'in the head' entails further actions to restore consistency". A simple example of repetition viscosity would be using a word processor to change the referencing style in a paper from a numbered list in order of appearance in the paper, to the Harvard style with an alphabetical list of references preferred for *Design Studies*. Here, a single conceptual change may result in many individual changes in the main text and to the references section. In knock-on viscosity, a small change results in a number of additional changes in order to return the system to a consistent state. Green & Blackwell give the example of adding a figure into a document. After the new figure is added, other figures may need to be renumbered, and indexes and cross-references within the text may need to be updated. Similar examples can be found in other digital designing tools. For example, in computer programming languages, renaming a function, a class or a method, or changing the 'visibility' of a method from public to private or protected, may result in knock-on viscosity as parts of the code that call the function or method must also be updated. Another dimension that has clear relevance to digital designing is *Visibility and Juxtaposability* which refers to the ease or difficulty of finding various parts of the notation, the ability to compare and combine parts, to view different parts of a project at the same time.

The dimensions are not completely independent of each other. To illustrate this point, consider *Abstraction*, which Green & Blackwell specifically relate to viscosity: "An abstraction is a class of entities, or a grouping of elements to be treated as one entity, either to lower the viscosity or to make the notation more like the user's conceptual structure."(ibid. p24). The previous example of updating the referencing style of a paper for a different journal can be managed by explicitly adding 'reference' as a new abstraction to a word processor. Tools such as Refworks (www.refworks.com) and BibTeX (Lamport, 1994) provide such abstractions. Similarly, if a word processor includes the abstraction 'paragraph style', then it may be possible to alter all of the 'headings' & 'sub-headings' without experiencing repetition viscosity. A number of other interactions between dimensions are discussed by Green & Blackwell.

Another cognitive dimension that may be familiar to users of digital designing tools is that of *Hidden Dependencies.* According to Green & Blackwell, a hidden dependency is "a relationship between two components such that one of them is dependent on the other, but that the dependency is not fully visible."[ibid. p17]. Examples of hidden dependencies include HTML links on the world-wide web, where it is possible to explore what a particular page is linked forwards to, but it is not possible to identify the set of all pages that link to a given page. Another example is in using a spreadsheet, where it is not possible to identify the set of cells that depend on the value held in a given cell. Similarly, in a computer-programming environment, it may or may not be possible to identify the set of all places from which a particular function is called.

It is important to notice that hidden dependencies are not necessarily undesirable. In using a spreadsheet package it is often helpful that the formulae that underlie the spreadsheet are (usually) hidden from view. Such observations reflect the fact that the cognitive dimensions framework is intended to support discussions of what might or might not be valuable in a particular system. Specific properties are not *a-priori* desirable or otherwise, and a property may be desirable for one part of a system but not another part. For example, in a word processor, it is quite beneficial that the style or font used for a word can be changed relatively simply reflecting a low viscosity, but changing the font and size used for an entire style (such as a heading or subheading) is more viscous. The cognitive dimensions framework provides a vocabulary with which the designers of tools for digital designing can reflect upon and discuss their design choices.

3.2 Cognitive dimensions and conversations using materials

Whilst some of the cognitive dimensions discussed above can be rooted in the immediate experience of conversing *with* digital materials, in the sense that viscosity and hidden dependencies reflect ways in which digital media might 'talk-back' to the digital designer, others can only be described with a contextual analysis of the situation in which the notation is used in creating utterances in dialogue with an audience, that is conversation *using* materials.

One dimension of particular interest for conversations using materials is *Secondary Notation*.

“Definition

Extra information carried by other means than the official syntax.

Redundant recoding gives a separate and easier channel for information that is already present in the official syntax. *Escape from formalism* allows extra information to be added, not present in the official syntax.

Thumbnail illustration

Indentation in programs (redundant recoding); grouping of control knobs by function (redundant recoding); annotation on diagrams (escape from formalism).” [ibid. p29, authors’ italics].

Secondary notation is commonly applied in digital devices. For example, adding free text comments to programme code is considered good practice with the aim being to benefit future programmers who may have to maintain the code. Modern office productivity software such as spreadsheets & word processors also offer opportunities to insert comments.

Other examples of dimensions related to conversation *using* materials include the dimensions of *closeness of mapping* and *provisionality*, which Green & Blackwell describe respectively as follows:

“**Definition**

Closeness of representation to domain

Thumbnail illustrations:

A close mapping: the visual programming language LabVIEW, designed for use by electronics engineers, is closely modelled on an actual circuit diagram, minimising the number of new concepts that need be learnt.” [ibid. p39]

“Definition

degree of commitment to actions or marks

Thumbnail illustration:

> Pencils are used by architects, typographers and other designers to make faint blurry marks, meaning 'something more or less like this goes more or less here', as well as precise hard marks." [ibid. p41].

In the context of the definition of material utterance above, the closeness of mapping of marks to the 'domain' of circuit diagrams referenced here, can only be understood in relation to a particular audience whose history includes familiarity with particular symbols for electronic engineering. Similarly, the notion of 'degree of commitment' to marks can only be understood in relation to a temporal account of a process of designing and development, in which the mark is used in an ongoing dialogue. Thus, whilst cognitive dimensions provides a pragmatic vocabulary for discussing the material properties of notational systems, the set of dimensions mixes concepts with different conceptual roots. As Blackwell & Green (1999) point out, a weakness of the cognitive dimensions framework is its lack of clear theoretical underpinning, making it difficult to establish whether the set of dimensions is complete and whether dimensions are orthogonal. The setting of designing as material utterance, with the associated issues of materials, genre and audience, may provide a basis by which such a systematic theoretical underpinning could be developed.

The cognitive dimensions framework has been applied to a wide range of notational systems. These include a number of tools that are used in different forms of digital designing. For example, Clarke & Becker (2003) describe how the technique has been applied by Microsoft to provide a predictive evaluation of the usability of software class libraries prior to release. Kutar *et al.* (2002) apply the approach to diagrams in the Unified Modelling Language (http://www.uml.org). Dearden *et al* (2003) use the technique to explore various approaches to prototyping software systems. Roast *et al*. (2004) apply the technique to support evaluation and redesign of a tool used by multimedia designers to create highly interactive Digital Versatile Disks (DVDs). The framework would appear to be a promising starting point for theorising about some aspects of digital designing in other domains.

4 Digital media as formal systems

In human-computer interaction Dix (1991) draws attention to the 'formality gap' between the messy world of human affairs and rigorously defined behaviour of computer systems. Dix and others argue that the complex (socially mediated) details of human interaction can never be completely captured by a closed formal system. For example, consider the privacy of personal information. I might prefer to keep my personal medical history private, but, if I was lying unconscious in the street in a foreign country, I would be happy for local doctors to access my personal details. I cannot to define or enumerate (that is, I cannot formalise) in advance, all the possible conditions under which I would or would not want my details to be available. Yet, for a software system to support such a function, the conditions under which information shall or shall not be available *must*, eventually, be rigorously defined. Martin *et al.* (1997) discuss the unending social and situational complexity that can be uncovered within the seemingly well-defined work of ambulance controllers. Ackerman (2000) describes the problem in terms of the 'social-technical gap'.

On the other hand, the formality of digital systems is a significant factor in accounting for their power. For example, the formal identification of pixel colours is essential to enable image manipulation software to select 'all the pixels of the same colour'. Formality also underlies the capability of digital systems to support reasoning about the consequences of design decisions, such as a computer aided design tool predicting the volume of a manufactured component, or calculating whether the planned layout of electrical and water services for a building is consistent with the structural design. Finally, the capability of digital systems to be replicated for minimal cost also relies on the formality of their structure.

4.1 Conversing across the formality gap

Formality, as described above, gives rise to an account of 'meaning' in computer science that is not situated, that is *finalized*. The 'meaning' of a computer system is treated as the behaviour of that system in response any given input, or some external definition of the response that should be expected. From this perspective, the challenge for software engineers constructing digital design tools, is to ensure that the relations executed by the

tool are internally consistent and are consistent with (finalized) external definitions of the expected behaviour. Formal mathematical techniques, including mathematical proof, are commonly used to support this activity, and subfields of computer science such as 'denotational semantics' (Allinson, 1987) study methods for reasoning about such correspondence. These properties correspond to what Goguen (1994) refers to as a 'representational theory of meaning' in which a sign represents something in 'the real world'. Representational theories of meaning are commonly adopted in the physical sciences. Within theoretical computer science, compositions of symbols in a computer programming language are interpreted as having a fixed representational correspondence to mathematical entities such as abstract algebras or calculi. These analytic tools have made significant contributions to the development of digital systems. The World Wide Web, XML and Java, all depend on insights that have been derived from this tradition. Goguen (1994) contrasts representational theories with social theories of meaning. Pragmatic accounts of 'utterance' (or material utterance), in which meaning is creatively constructed by speaker and audience in context, clearly belong in this latter category. Gogeun argues that both theories of meaning must be accommodated to support the successful designing of digital systems.

The inclusion of speech genre within our account of material utterance, permits a theoretical resolution. In Bakhtin's view, a speech genre is a way of seeing the world, that is learned socially through interaction. Thus these alternative theories of meaning can be regarded as belonging to different speech genres. From a social perspective, the 'challenge for software engineers' noted above '… to ensure that the relations executed by the tool are internally consistent and are consistent with (finalized) external definitions of the expected behaviour', is part of the social field of answerability associated with the material utterance of releasing a digital designing tool. The genre in which design tools are released includes the expectation that tool producers are accountable to users for explaining and maintaining the relationships between inputs and outputs. Broader social mechanisms such as standards bodies, licensing agreements and legal systems are also relevant. Here, a representational theory of meaning is used instrumentally to create the tool and to help the tool authors to discharge their social responsibilities to users. As

MacKenzie (2001, 2004) shows, even such apparently clear-cut representational activities as mathematical proof can be usefully analysed as social processes.
This resolution of the social and representational theories of meaning implies that a prototype or specification developed using a digital designing tool might be understood as having a representational relationship to some world of possible systems, but its presentation (for example to a group of clients, users or other designers) is a material utterance where the 'meaning' must be sought in the pragmatic social context.

4.2 Formality and genre

In order for a notation to operate successfully in coordinating designing activity, there is a need for a social network of people to adopt compatible understandings of the relationships between configurations of symbols in the notation, and expected responses. Introducing a new tool for designing is an intervention in an established system of artefacts, practices and interpretations. Spinuzzi & Zachry (2000) characterise such systems in terms of 'genre ecologies'. As Coyne et al. (2002) observe, new digital devices introduce "new practices, new terms and metaphors …new work patterns and practices … new forms of organisation, new specialisms …" (ibid., p271). Successful adoption of a new tool then depends on finding a successful niche in relation to the established genre ecology. As with any ecology, if a new device finds such a niche, it may displace or transform related devices and activities as the genre ecology evolves. Blandford et al. (2002) have proposed a way of exploring tensions between the representational theories that reflect the formality of digital tools and the genres familiar to practitioners for whom the tools are intended. Their technique of 'Ontological Sketch Modelling' may be particularly appropriate to domains of knowledge work such as digital designing. In Blandford et al.'s approach, tool designers create a hierarchical map that details entities, properties and relationships that are discussed by the practitioners who are the intended users of a system. A similar map is created to show the entities, properties and relationships that are apparent from the interactive structure of the proposed system. Such analyses can be understood as investigating one aspect of the genres that provide the context into which the new tool is to be introduced. Ontological analysis focuses on areas of match and mismatch between the two maps. The assumption

is that, if the maps are well matched, then the software will be more readily accommodated into the existing genre ecology of the intended users.
In ontological sketch modelling, the maps can be viewed as applying both theories of meaning: the map of practitioners' speech genre being understood socially, whilst the map of the software could be understood representationally. Viewing the design approach as a whole, the maps themselves support material utterances that allow software designers and intended users to explore key aspects of the software (that may be difficult to modify later in the development cycle) and possible implications for the future evolution of the genre ecology.

4.3 Formality and toolkits

Many digital designing systems are distributed in a way that provides not only a mechanism for interpreting a defined input language, but also includes a collection of pre-defined, re-usable components that are consistent with the language. Examples may vary from the 'clip-art' that is distributed with office presentation software, through to large-scale software development libraries such as the Java SDK, or J2SE. One way to characterise these types of distribution is as a 'toolkit'. The toolkit metaphor draws attention to the way that such distributions change the stock of material available to a speaker in constructing the material elements of an utterance. Doorst & Cross (2001) describe how designers take into account their available resources in developing both their understanding of the 'problem space' and 'solution space'. One possible effect of this may be a tendency to devise solutions that are readily supported by components of the toolkit, with the potential consequence of limiting creative development of novel solutions. Given the ease with which digital devices are replicated, this is a potentially strong effect. On the other hand, as Coyne *et al.* (2002) point out, digital designing tools may permit artefacts to be designed that may have been unachievable using non-digital media. Using representational theories of meaning in digital systems also makes possible techniques such as finite element analysis, required to verify the structural properties of non-rectilinear structures such as the Guggenheim museum in Bilbao. Here, the new capabilities allow the creation of an utterance that extends the genres of artefacts that are the results of designing activities. Which of these two tendencies will dominate may be

dependent on the effort made by designers to explore the full range of possibilities available in digital media, and on the way that utterances that illustrate novel possibilities gradually extend the genres that designers encounter.

5 Tools & practices of digital designing

In the previous sections, I have reviewed two key areas of work in human-computer interaction and the psychology of programming, and their potential lessons for theories of digital designing. In this section I explore some consequences of these accounts for the users and makers of tools for digital designing.

5.1 Responses to materials in digital designing

As noted above, the cognitive dimensions framework provides an initial vocabulary for discussing the experience of conversing with materials in digital designing. Considerations such as *viscosity*, *hidden dependencies* and *abstraction,* have been shown to be useful analytic tools when examining a range of different designing environment. However, the majority of the applications of the framework to date have been in the context of software development, in which the 'notation' selected is either a programming language, or a software modelling language such as UML.
Dearden *et al.* (2003) show how some elements of the framework could be applied to tools used in the early prototyping of software systems, including paper-prototyping as well as recent hybrid tools such as DENIM (Newman et al. 2003) that seek to achieve some of the advantages of paper-prototyping in the context of a software-based prototyping environment. In DENIM a web designer uses a graphics tablet and a pen to sketch web pages and to record links between pages (see figure 3). The resulting website design can then be executed in a specially designed web-browser. This approach aims to retain some of the low viscosity and high degrees of provisionality associated with paper-prototyping, whilst exploiting the formal capabilities of the digital medium to allow a user to see a working prototype at a very early stage of the designing process.

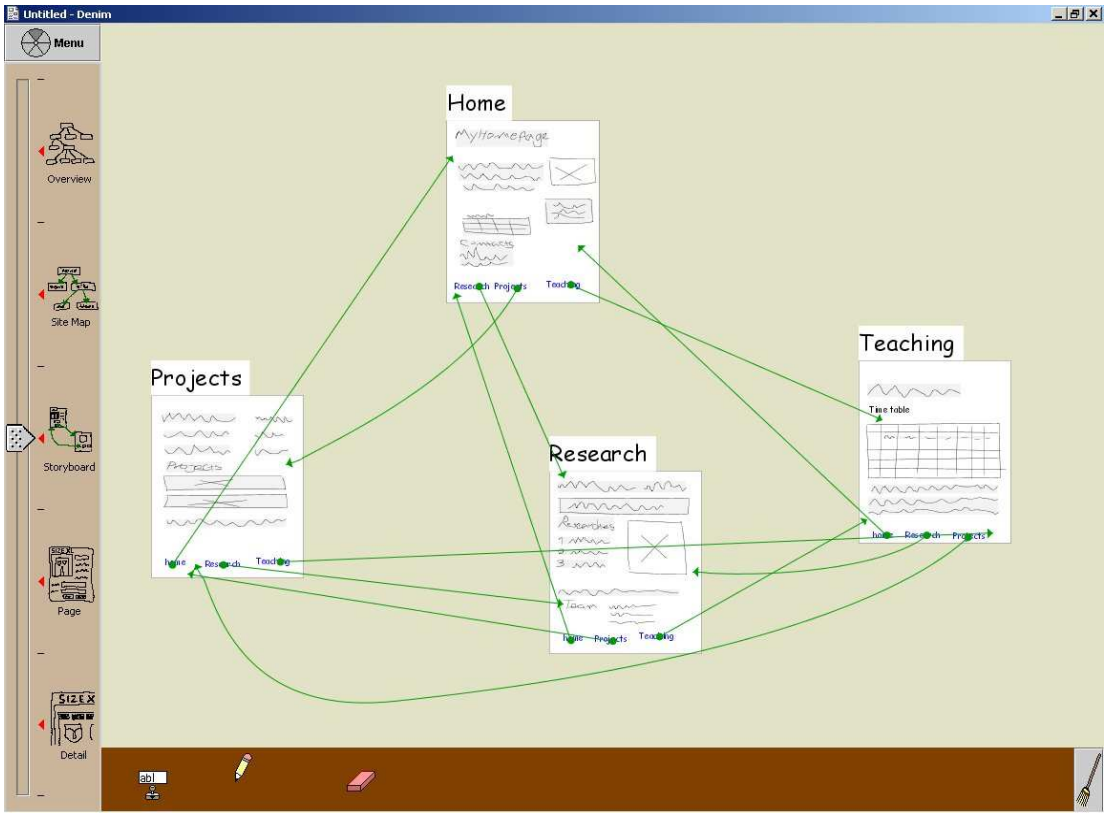

Figure 3: The DENIM website design environment (see Newman et al., 2003).

However, Dearden et al.'s (2003) analysis indicates some areas of possible incompleteness and ambiguity within the cognitive dimensions framework in relation to prototyping. For example, the dimension of *Provisionality* may be further broken down by considering whether a mark is being used as provisional in the context of the design process, i.e. the mark has a specific agreed relationship to some object in the design domain, but the utterance within which the mark is used highlights the fact that it is intended only as a 'placeholder'; or whether the mark is being interpreted as ambiguous in itself (or ambiguous in relation to the genres surrounding the designing). This finding suggests that exploring other domains of digital designing using the cognitive dimensions framework may highlight other ways in which the framework might be usefully developed and enhanced.

5.2 Responses to audience in digital designing

Given that digital designing tools exist within social processes of dialogue, it becomes useful to consider the relationships between these tools and these social processes. As

indicated earlier, an important cognitive dimension for such social processes is the availability of ‘secondary notations’. By allowing secondary notations, a highly formal language (such as a computer programming language or a computer aided design environment) can include the possibility of adding supplementary comments that fall outside the normal syntactic constraints of the tool. Green & Blackwell discuss this as “escape from formalism” (Green & Blackwell, 1998, p29). For example, most computer programming languages provide a means for including free text comments within the code, which may be used to explain the structure of the code to future programmers responsible for maintenance. Similarly, modern word processing or spreadsheet software permits users to add, modify or delete comments that are in some sense ‘secondary’ to the main notation. In designing, secondary notation is commonly used to support utterances about the designing process or about the status of existing proposals.

Dearden *et al* (2004) describe the introduction of a secondary notation to the DENIM environment (Newman et al., 2003). Gabbeh (Dearden et al., 2004; Naghsh & Ozcan, 2004) extends DENIM by allowing a web designer to associate a comment with a page or parts of a page. A novel feature of Gabbeh is the facility to add comments to the design. The system also allows a user to add comments as the design is being executed or demonstrated (see figures 4 and 5). These extensions reflect a change in the focus of the tool, making the representational relationship between the sketches and the finished website less central, instead using the tool to support both designers and users communicating through material utterances.

Boujut & Laureillard (2002) describe how a computer aided design environment was extended with specific notational conventions to allow designers responsible for the form of the product, and designers responsible for the production process to exchange utterances so that production constraints could be properly taken into account. By introducing such conventions, Boujut & Laureillard’s proposals will impact on the whole practice of designing, and the genres of dialogue in the designing activity.

These systems suggest two ways in which tools for digital designing can support the role of audience in material utterances. Other possibilities will no doubt be developed in future.

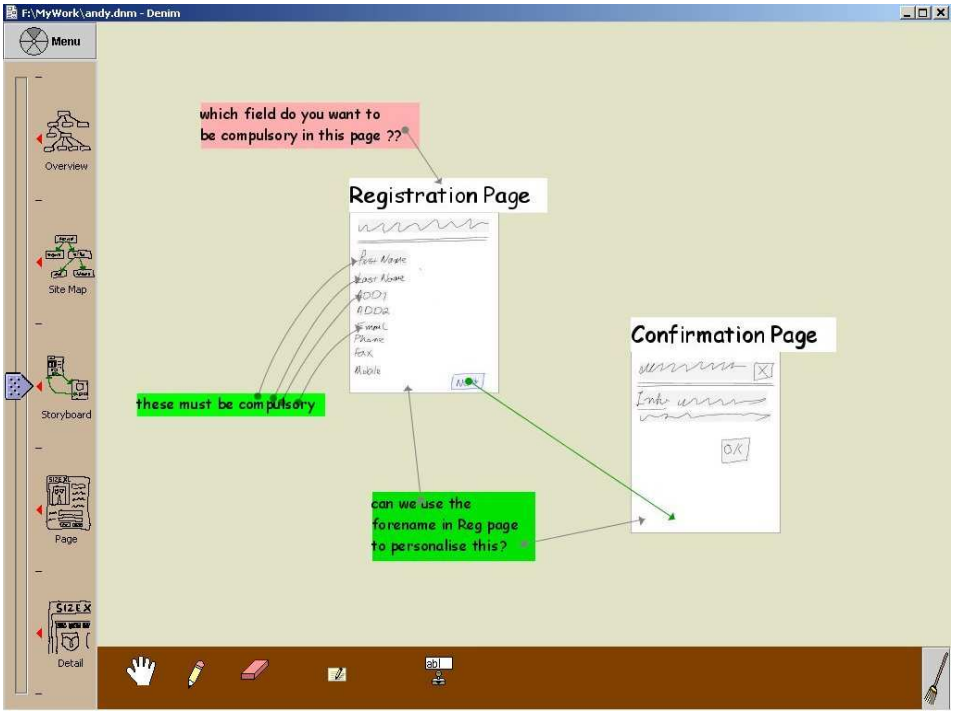

Figure 4: Gabbeh: design view

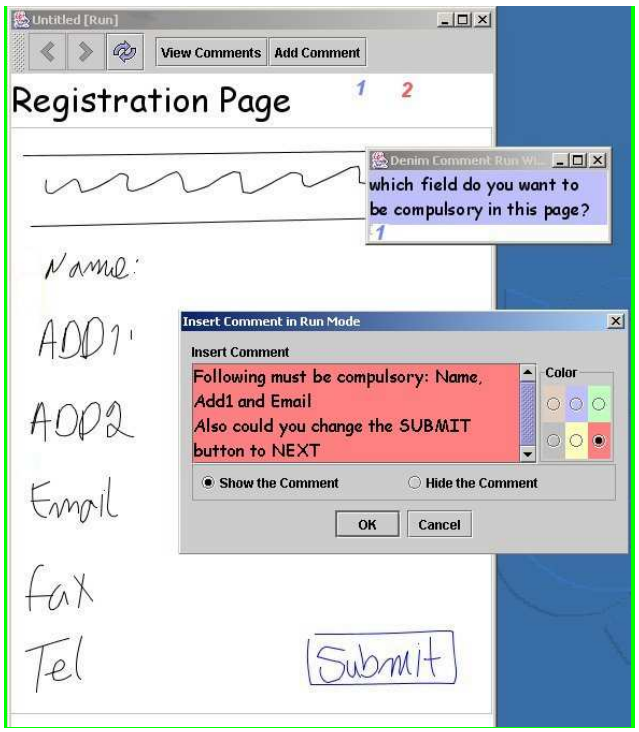

Figure 5: Gabbeh: adding comments at run time

5.3 Responses to genre in digital designing

Roast et al. (2004) examine the case of DVD Extra Studio, a digital designing workbench for producing highly interactive DVDs (Digital Versatile Disks). DVD Extra Studio allows authors to select and specify a procedure that can 'generate' literally thousands of nodes and links onto a DVD, thus opening the possibility of far richer viewer interaction with an ordinary DVD player, including dynamic game playing. Existing examples produced with the technology include an implementation of the TV quiz game 'Who wants to be a millionaire?' (Universal Pictures, 2003) as well as adventure games including more than 2^{30} discrete states. However the technology also challenges the established genres of DVD designing, by introducing complex programming activities into the work.

Roast et al.'s analysis of DVD Extra Studio applies both the cognitive dimensions framework of Green & Blackwell (1998) and the Ontological Sketch Modelling approach of Blandford et al. (2002). The complex setting of an innovation such as DVD Extra Studio challenges the Ontological Sketch Modelling technique because the analysis must consider four different genre ecologies as relevant to the tool. the relevant genres were: the structures that were named and represented directly by the user interface of the workbench; the technical specification of DVD formats including terms such as 'cells', 'IFO files', and 'sub-pictures'; the established world of DVD production using traditional tools; and finally the world of film and TV production.

The analysis shows how a single term may be present in multiple genres, but may refer to entities that have important differences. For example, the term 'storyboard' is common to: TV and film editing, where it refers to a linear presentation of the intended structure of a production; traditional DVD editing where it is extended to provide a branching structure; and to DVD Extra Studio where much richer relations might exist between parts of a storyboard. Setting up metaphors of this kind is a common approach to interaction design and can help users to make sense of new technologies, thus helping them to integrate them into their practices. On the other hand, where major differences occur between the existing genres and the new technology, the expectations of the users drawn from previous genres may actually make uptake more difficult.

Other parts of the analysis show the value of drawing on the existing genres to identify important elements to include within a digital designing tool. For example to create navigational branches, in the early version of DVD Extra Studio that was analysed, involved editing both an underlying image and a separate 'sub-picture' that displays a highlight over the currently 'selected' option. This arrangement reflects the underlying technological implementation in the DVD specification. For users of the tool (i.e. DVD producers), this means that when a change of appearance is required, both the image and the sub-picture must be edited as separate objects using an external graphic editor, and then the alignment must be checked. However, in the genre of traditional DVD production and viewing, such choice points are usually referred to as 'menus'. Some existing DVD editing software supports the concept of menu directly, allowing a DVD producer to specify that a menu is based on a layered image file, with different states of the menu corresponding to different combinations of layers being displayed or hidden. Holding all these different components of the menu in a single image file greatly simplifies this part of the editing task. Here, the analysis of the genre of designing activity provides useful clues to elements that can be included in DVD Extra Studio to ease its adoption within the existing genre ecology.

Roast et al.'s analysis suggests that direct ease of use (conversation with materials) provides only a partial account of factors affecting the uptake of this digital designing tool, with issues of genre acquiring at least an equal weight. Yet, genre is not here seen as a passive constraint, rather it is seen as a source for design ideas. Indeed, the utterance of releasing a tool such as DVD Extra Studio will itself impact upon the genre associated with DVD production and viewing by offering new interactive possibilities.

6 Conclusions

In this paper, I developed a framework for considering designing activity in terms of the concept of a material utterance. I proposed a working definition of designing activity as:

> the making of utterances (material or otherwise) that are oriented towards the form of future utterances to involve a different audience.

This view of designing suggests that if digital designing differs from other forms of designing, then accounts for the differences should be sought in the material properties of

digital systems and the genres of work practice that surround their use. Previous work in Human Computer Interaction and the Psychology of Programming provides useful tools for thinking about such material properties and existing genres. Reflecting on: the nature of conversation *with* digital materials; the possibilities for conversing with an audience *using* materials; and the *genres* that provide context for utterances; provides a rich source of insights to inform the development of tools for digital designing.

On the other hand, the account of designing in terms of material utterance provides challenges to this previous work and suggests that careful examination of digital designing could be used to strengthen and refine these analytic techniques.

7 Acknowledgements

The work in this paper has been supported by EPSRC Research Project Grant number: GR/R87918 paperCHASTE. I would like to acknowledge the help & advice I have received in developing this paper from Peter Wright and John McCarthy, particularly with regard to readings of Bakhtin. Also to my colleagues Chris Roast, Chris Rust & Simeon Yates for helpful comments on earlier drafts of the paper.

References

Ackerman, M., 2000. The Intellectual Challenge of CSCW: The Gap between Social Requirements and Technical Feasibility. *Human Computer Interaction* 15(2/3).

Allison, L. 1987. *A Practical Introduction to Denotational Semantics.* Cambridge University Press.

Austin, J.L., 1962. *How to do things with words.* Oxford University Press.

Bakhtin, M.M., 1994. *Speech Genres & Other Late Essays.* Translated by V.W. McGee, Edited by Emerson, C. & Holquist, M. University of Texas Press, Austin, Texas.

Blackwell, A.F. & Green, T.R.G., 1999. Investment of Attention as an Analytic Approach to Cognitive Dimensions. In T. Green, R. Abdullah & P. Brna (Eds.) *Collected Papers of the 11th Annual Workshop of the Psychology of Programming Interest Group (PPIG-11)*, pp. 24-35.

Blackwell, A.F. & Green, T.R.G., 2000. A Cognitive Dimensions questionnaire optimised for users. In A.F. Blackwell & E. Bilotta (Eds.) *Proceedings of the Twelfth Annual Meeting of the Psychology of Programming Interest Group* , pp 137-152.

Blandford, A., Wong, B.L.W., Connell, I. & Green, T., 2002. Multiple Viewpoints On Computer Supported Team Work: A Case Study On Ambulance Despatch. In Faulkner, X., Finlay, J. & Detienne, F. (Eds.) *People and Computers XVI – Memorable Yet Invisible, Proceedings of HCI 2002*. Springer, pp. 139 – 156.

Boujut J.F. & Laureillard, P. 2002. A co-operation framework for product-process integration in engineering design. *Design Studies* 23, pp 497 – 513.

Bowers, J. & Pycock, J., 1994. Talking Through Design: Requirements and Resistance in Cooperative Prototyping. *In Proceedings of CHI '94*, Boston, MA. ACM Press, pp 299 – 305.

Bucciarelli, L. L., 2002. Between thought and object in engineering design. *Design Studies* 23. pp. 219 – 231.

Carroll, J.M., Kellog, W.A. and Rosson, N.B., 1991. The Task - Artifact Cycle, in J.M.Carroll (Ed.) *Designing Interaction - Psychology at the Human-Computer Interface*, Cambridge University Press, pp.74-102.

Clarke, S. & Becker, C., 2003. Using the cognitive dimensions framework to measure the usability of a class library. Presented at the 15th *Annual workshop of the Psychology of Programming Interest Group*, Keele University, UK, 8th –10th April, 2003.

Coyne, R., Park, H. & Wiszniewski, D. 2002. Design devices: digital drawing and the pursuit of difference. *Design Studies* **23,** 263 – 286.

Dearden, A.M., Siddiqi, J. & Naghsh, A., 2003. Using Cognitive Dimensions to Compare Prototyping Notations. Presented at the *15th Annual Meeting of the Psychology of Programming Interest Group*. Keele University, UK, 8th –10th April 2003. Available at: http://www.shu.ac.uk/schools/cms/paperchaste/downloads/publications/dsn2003.pdf

Dearden, A., Naghsh, A. & Ozcan, M. 2004. Support for participation in electronic paper prototyping. In *Proceedings of the eighth Participatory Design Conference, 2004*, volume 2. ISBN 0-9667818-3-X. CPSR, Palo Alto, CA, pp 105 – 109.

Dix, A. (1991) *Formal methods for interactive systems.* Academic Press.

Doorst, K. & Cross, N., 2001. Creativity in the design process: co-evolution of problem-solution. *Design Studies* 22, pp. 425 – 437.

Fish, S. 1980. *Is There a Text in This Class? The Authority of Interpretive Communities.* Cambridge, Mass.: Harvard Univ. Press.

Goguen, J.A. 1994. Requirements engineering as the reconciliation of social and technical issues. In Jirotka, M. & Goguen, J.A. (Eds.) *Requirements Engineering Social and Technical Issues*, pp.165 – 200. Academic Press, London.

Green, T.R.G. and Petre, M., 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *J. Visual Languages and Computing*, 7, 131-174.

Green T. & Blackwell, A., 1998. Cognitive Dimensions of Information Artefacts: a tutorial. Available from http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/CDtutorial.pdf

Hutchby, I. and Wooffitt, R. 1998. *Conversation Analysis: principles, practices and applications.* Cambridge: Polity Press.

Kac, E., 1999. Negotiating meaning: the dialogic imagination in electronic art. In *Proceedings of Computers in Art and Design Education Conference*, University of Teesside, UK, ISBN 0-907550-66-5.

Kapor, M., 1991. A Software Design Manifesto.: Time for a Change, *Dr Dobbs Journal* 172 (January, 1991), 62 – 68. Reprinted in Winograd, T. (Ed.), 1996, *Bringing Design to Software*, Addison Wesley, Reading MA., pp 1 – 9.

Kerry, A. (Ed.), 1994. *The Wittgenstein Reader.* Blackwell. ISBN 0-631-19362-6.

Kutar, M., Britton, C. and Barker, T., 2002. A Comparison of Empirical Study and Cognitive Dimensions Analysis in the Evaluation of UML Diagrams. In J. Kuljis, L. Baldwin and R. Scoble (Eds), *Proceedings of the Fourteenth Annual Meeting of the Psychology of Programming Interest Group*, pp. 1-14.

Lamport, L. 1994. *LaTeX: A Document Preparation System 2nd Edition*. Addison Wesley. ISBN: 0-201-52983-1

Levinson, S. 1983 *Pragmatics* Cambridge: Cambridge University Press

Liddament, T., 1999. The computationalist paradigm in design research. *Design Studies* 20, pp. 41 – 56.

Lloyd, P., 2000. Storytelling and the development of discourse in the engineering design process. *Design Studies*, 21, pp. 357 – 373.

Lloyd, P. and Snelders, D. What was Philippe Starck thinking of? *Design Studies*, 24, pp. 237 – 253.

MacKenzie, D. 2001. *Mechanizing proof: computing risk, and trust.* Cambridge, MA: MIT Press.

MacKenzie, D. 2004. *Computing and the cultures of proving*. Paper for Royal Society Discussion Meeting on the Nature of Mathematical Proof, London, 18-19 October, 2004. Available from: http://www.sps.ed.ac.uk/staff/Royal%20Society%20paper.pdf

Martin, D., Bowers, J. &Wastell, D., 1997. The interactional affordances of technology: an ethnography of human computer interaction in an ambulance control centre. In Thimbleby, H., O'Conaill, B., & Thomas, P. (Eds.) People and Computers XII. *Proceedings of HCI 1997*, pp. 263-282. London: Springer.

McCarthy, J & Wright, P., 2004. *Technology as Experience*, MIT Press, Cambridge, Mass.

Morris P. (Ed.), 1994. *The Bakhtin Reader* Edward Arnold, London.

Morson, G. S. & Emerson, C., 1990. *Mikhail Bakhtin Creation of a Prosaics*. Stanford University Press, Stanford, CA.

Naghsh, A. & Ozcan, M., 2004. Gabbeh:- a tool for computer supported collaboration in electronic paper prototyping. In Dearden, A & Watts, L. (Eds). *Proceedings of HCI 2004: Design for Life Volume 2*. pp 77 - 80: British HCI Group. ISBN: 1-897851-13-8.

Newman, M.W., Lin, J., Hong J.I. & Landay, J.A., 2003. DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice, *Human-Computer Interaction* 18(3): pp. 259-324.

Roast, C., Dearden, A., & Khazaei, B., 2004. Enhancing contextual analysis to support the design of development tools In Fincher, S., Markopoulos, P, Moore, D. & Ruddle, R. (Eds), *People & Computerse XVII: Design for Life, Proceedings of HCI 2004*. Springer.

Rockwell, G., 2002. Gore Galore: Literary Theory & Computer Games. *Computers and the Humanities* **36**: pp 345 – 358.

Schön, D., 1995. *The Reflective Practitioner How Professionals Think in Action*. Alershot, UK: Arena.

Schon D. & Bennet J., 1996. Reflective Conversation with Materials. In Winograd, T. (Ed.) *Bringing Design to Software*. Addison Wesley, pp 171 - 184.

Searle, J. 1969. *Speech Acts: an essay in the philosophy of language*. Cambridge University Press.

Spinuzzi, C., 1999. Grappling with distributed usability: A cultural-historical examination of documentation genres over four decades. *Proceedings of the 17th annual international conference on Computer documentation*. ACM Press. pp. 16 – 21.

Spinuzzi, C. & Zachry, M., 2000. Genre Ecologies: An Open-System Approach to Understanding and Constructing Documentation. *ACM Journal of Computer Documentation* 24, pp. 169 – 181.

Universal Pictures, 2003. *Who Wants to be a Millionaire, Interactive DVD*. Universal Pictures Video.

Woolgar, S., 1994. Rethinking requirements analysis: Some implications of recent research into producer-consumer relationships in IT development. In Jirotka, M. & Goguen, J. A. (Eds.) *Requirements Engineering Social and Technical Issues*, pp.201 – 216. Academic Press, London.

Wright, P. & McCarthy, J., 2005. The value of the novel in designing for experience. In, Pirhonen, A.; Isomäki, H.; Roast, C. & Saariluoma, P. (Eds.) *Future Interaction Design*. Springer, pp. 9 – 30. ISBN: 1-85233-791-5.

Yates, S., Taylor, S., Wetherell, M., 2001, *Discourse as data: A guide for analysis*, London; Sage.