Searching for methodology: Feminist technology design in computer science

Corinna Bath
Stadlgasse 2/5r
8020 Graz
bath@sts.tugraz.at

ABSTRACT

This paper introduces a systematic feminist approach to conceptualizing and building computational artefacts. The main objective is to provide methods for technological design that avoid a perpetuation of the existing structural-symbolic gender order. This, however, presupposes a thorough theorizing and analysis of gendering processes.

Based on a review of existing research on ‘gender in information technology’ I will describe four mechanisms that often lead to gendered computational artefacts: 1.) the ‘I-methodology’ that assumes technology as neutral, 2.) implicit gendered assumptions and the gendered distribution of labour, which are inscribed into computational artefacts, 3.) gender stereotypes of human bodies and behaviour reflected in technology and 4.) de-contextualization and disputable epistemological and ontological assumptions. For each of these mechanisms I will propose technology design methods adopted from the field of ‘critical computing’ and discus their potentials and limits to de-gender computational artefacts on the basis of feminist theory.

Keywords

De-gendering of computational artifacts, technology design methods, feminist theory

INTRODUCTION

‘Gender studies in computer science’ is a new research field, which was only recently established at German universities.[1] Most studies in this area are considered to address the problem of getting more women into computer science and IT professions.[2] Recently a second approach, which focuses on alleged gender differences in the use of technologies, gains public attention. According to this line of argument, research and development should take into account that women use technology differently from men (cp. e.g. [19], [55]).

The first view reduces gender to a women’s problem. Women should adopt existing structures and technology. In drawing on traditional ideals of science as rational, value-free project it assumes technology to be (gender) neutral and, therefore, does not question practices of technology design and development. The second view aims to change technology by “discovering” women as a new consumer target group. Since new features are implemented into technology meant and promoted “for the woman”, this approach tends to essentialize gender.[3] Both understandings seem to ignore the wide corpus of knowledge on gender and technology that feminist researchers in gender studies, computer science and in science and technology studies developed during the last three decades (cp. e.g. [31], [49], [6], [39], [57], [5], [74]). They do neither recognize the failure of purely “women into technology!” initiatives nor do they acknowledge the numerous empirical insights into the complex entanglement of gender and computer science.

This contribution aims to overcome the shortcomings of public discourses on gender and IT, but also some of those in existing research about this area. First, it shifts the focus from women to the technology. Along with the second view computational artefacts should be altered rather than women or female users. However, the envisioned change cannot be built upon essentialism. Instead of assuming fundamental gender differences technology should avoid a reproduction and further stabilization of the existing structural-symbolic gender order. Thus, the aim is counteract problematic inscriptions of gender into technology. I call this objective ‘de-gendering of technology’ [4]

[1] Evidence for this process of institutionalization can be seen in new professorships for gender studies in computer science departments (since 1998 at Bremen University, since 2004 at Hamburg University).

[2] See e.g. the recent National Pact “Komm, mach MINT” by the German Government in June 2008 that aims to recruit more girls and young women in math, computer science, science and engineering. http://www.komm-mach-mint.de/Startseite/Nationaler-Pakt/Memorandum.

[3] The study ‘Discover Gender’ [19] at the Fraunhofer research company demonstrates this view prominently. Another example of such a direction is the Volvo concept car, see e.g [66].

[4] This terminology implies that there cannot be a gender-free zone.

In order to accomplish this goal, the gendering of technologies is, second, rather located in the process of conceptualization and building artefacts and not mainly in the end product. Although, gendering processes become manifest in products. The approach I will introduce in this paper focuses on methods to design technology alternatively. Gearing towards alternatives means an intervention into traditional practices in computer science. Allies for such a project can be found in the field of "critical computing"5, which already provides several approaches to integrate users perspectives, critical reflection and social theory into technology design. It is, however, the question, which methods might support a de-gendering process in which case of technology.

The question, which technology design method can be used as a promising de-gendering strategy for a certain technology, points at research lacunae of gender studies in computer science. So far only participatory design methods were proved successful to develop software at women's work places (cp. e.g. [69], [56], [33]). Furthermore, a few guidelines appeared recently that recommend a gender sensitive design (cp. [19]). However, there has been no basic approach yet that addresses the question systematically. A theoretical foundation and a close analysis of the gendering of computational artefacts is a prerequisite for a methodological de-gendering approach. This indicates the need for clarification in which way computational artefacts can be gendered.

In the following I will summarize some main results of my dissertation thesis that addresses theoretical aspects of the gendering of computational artefacts, includes a systematic analysis of case studies and proposes technology design methods that might be called feminist. I start with a brief description of the theoretical background and then proceed with a characterization of four mechanisms that often lead to gendered computational artefacts. For each of these mechanisms I will propose technology design methods adopted from the field of "critical computing" and discus their potentials and limits to de-gender computational artefacts on the basis of feminist theory.

THEORIZING THE GENDERING OF COMPUTATIONAL ARTEFACTS

Rather than seeing gender as an essential attribute of women and men or gendering processes as intentional inscription into technologies targeting at discrimination, technologies are conceived to have gender politics (cp.[73]). Since the computer science discipline does not provide any insights and methods for this research perspective I mainly draw on approaches from the field of science, technology and society. Actor-network theory (cp. e.g. [40], [41], [42]) seems to be a good starting point to understand the relationship between technology, gender and society in its complexity. However, this approach was criticized by feminist scholars (for an overview cp. [72], [62], see also [25]). So I prefer to draw on the work of feminist technoscience researchers, particularly Donna Haraway [35], [36], Karen Barad [8] and Lucy Suchman [64]. Just as actor-network theory these three authors do not assume that there are essential differences between humans and technical artefacts. They propose a theoretical framework which allows thinking the configuration of humans and machines in heterogeneous networks or as hybrids—without loosing out of sight the political character of technological artefacts. Most notably, they aim at accountability in research & development that might include an alternative design of technology.

As opposed to Latour [40], Barad claims that the relation between humans and non-humans is asymmetric. Furthermore, she calls on feminists to acknowledge materiality. On this basis Suchman considers not only agency, but also accountability as distributed between humans and machines. She transfers the theoretical approaches of feminist technoscience research to computer science and, thus, opens up the horizon to conceptualize the gendering and a de-gendering of computational artefacts.

To capture the concrete processes of gendering of technology, the concept of gender scripts (cp. [50], [51], [54] referring to [3]) is well-known. In focusing on user representations, however, gender scripts cannot explain the gendering of other computational artefacts such as concepts of humanness used in AI (cp. [64], [11]), modelling methods like object orientation (cp. [23]) or classifications schemes (cp. [18]) and dichotomies (cp. [70], [11]) that underlie technology design.[6] All these artefacts, though, have been proved to have (gender) politics. To understand this kind of gendering of artefacts, Barad's concept of 'posthumanist performativity' [7] is a helpful tool. Posthumanist performativity is—as she describes—a materialist and posthumanist reworking of Judith Butler's notion of performativity ([20], [21]). This notion, according to which sex is an always contested materialization of gender norms, was developed within the context of feminist theorizing of the body. Barad, however, criticizes that Butler understands matter (i.e. the body) as a passive product of discursive practices rather than conceiving it as an active agent participating in the process of materialization. Posthumanist performativity, moreover, is "a robust account of the materialization of all bodies—'human' and 'non-human'—and the material-discursive practices by which their differential constitutions are marked" ([7], 810). On this basis the gendering of artefacts can be conceptualized as a co-materialization of matter (or computational artefacts, respectively) and gender. This concept allows describing the gendering of software applications, information systems and user interfaces as

[5] See the decennial conferences on 'critical computing' in Scandinavia (e.g. [14]), and the 'Theorie der Informatik' conferences in Germany (e.g.[47]).

[6] See [10] for a discussion of the potentials and limit of the gender script concept

well as the gendering ofconcepts and assumptions in technology design, modelling methods and basic research.

DE-GENDERING OF COMPUTATIONAL ARTEFACTS AS A SITUATED APPROACH

The next step to address the question of how to design computational artefacts that might be characterized as 'de-gendered' is to determine a goal, i.e. an envisioned outcome that a de-gendering (design) process should achieve. Starting from feminist theory and feminist STS, there are several options as to what a de-gendered design of information technologies could mean: Should technology designers acknowledge differences between users (i.e. gender difference approach), in order to provide equal access? Should technology support competences assigned to the female realm as much as those traditionally considered to be masculine (i.e. equity approach)? Or should technology enable users to question the existing binary sex and gender system (i.e. deconstruction approach)? In this paper, I propose a situated approach, in which the de-gendering strategy to be chosen is based on a close analysis of the technological artefacts in question. As a rough guideline I distinguish four processes how computational artefacts might be gendered—a taxonomy that was derived from a systematic analysis of case studies in feminist STS and computer science. In the following four sections I will describe these four mechanisms and suggest certain technology design methods that seem promising for a de-gendering process.

Alleged neutral technology and the 'I-methodology'

Designers assume many technologies to be neutral, but a closer analysis reveals barriers in use. One example of these kinds of artefacts are early speech recognition systems in Artificial Intelligence that were not able to recognize female voices, since the designers did not think about the fact that adapting the technology to male voices could exclude female users (cp.[19], 6). Another case study, however, shows that even if designers explicitly aim to build technology 'for everyone' they are still in danger of excluding certain users by design. Els Rommes unmasked the development of the Digital City Amsterdam as a design for hegemonic masculine interests (cp. [52], [54]). She discovered that designers undermined their own agenda 'XS4all' (speak: access for all) by using the 'I-methodology'—a form of implicit user representation. They unconsciously assumed that users would have the same technical equipment, knowledge und skills, the same preferences and interests, and thus, see themselves as representatives of the users. Since they often form a homosocial, predominantly masculine group they actually inscribe their own background, knowledge, concerns and attitudes into the technology.

While early speech recognition systems could not be used by women, because of their higher pitched voice, i.e. a more or less biological gender difference, the designers of the Digital City Amsterdam did not question socio-economic prerequisites, such as access to the latest generation of computers, some experience in the use of the internet and the trial-and-error strategy. As Anne-Jorunne Berg [13] pointed out, such a structural exclusion of women and other 'others' from the use of certain technologies can already occur on the level of problem definitions that underlie technological solutions. Her study of three prototypes of 'smart houses' illustrates that the designers were not aware of housework, which is traditionally assigned to the female realm. Rather, they implicitly assume the customer to be a man interested in technology, not unlike the stereotype of the computer nerd.

All these examples illustrate the need for technology design methodologies that take into account a variety of users. Hence, the objective when facing alleged neutral technologies is the inclusion of diverse users, equal access and usability. The methodologies sought after should strive for the acknowledgement of differences, e.g. physical and social gender differences, but also cultural, class, age and other categories.

Several subfields of computer science already have a long tradition of developing methods of technology design that aim to avoid the I-methodology. Ergonomics, socio-technical systems design and human-computer interaction focus on getting to 'know the user' (cp. [34]), in order to build technologies for use and the real user instead of expecting that users will adapt to already existing technology (cp. e.g. [24], [43], [48]). In the cases of the Digital City Amsterdam and the smart houses, designers could have conducted usability tests to realize that their products do not match the skills, interests and preferences of the envisioned target group of the technology (cp. [52]).. An alternative design, however, should start with a thorough requirements analysis of the intended users—not in the sense of requirements engineering, but understood as part of an evolutionary or cyclic user-centred design (cp. e.g. [15], [53]). Although it has to be discussed which representatives of the users should be chosen, if the technology is meant to be used by everyone, particularly involving diverse, e.g. female users in the design process seems to be a way of preventing technologists from the mistakes of the I-methodology.

Technologies for 'the female user', stereotypes and the gendered division of labour

A second class of technologies contains those which are built for specific users, e.g. women as customers or to support women in their workplaces, but which in effect codify gender difference and reinforce the traditional gender hierarchy. Examples of this kind are the round dialogue box for font selection designed by the graphic designer Aaron Marcus [45] for white American women, which is built upon the assumption that females would prefer curvilinear shapes, or the early word processing software Jeanette Hofmann [37] analyzed, which assumed secretaries to be permanent beginners and by design defined them as technically unskilled users. Other case studies, for instance in the fields of nursing and call-centre service work (e.g. [44], [69]), show the lack of knowledge on ‘invisible work’ since these software systems were modelled in a way that fails to adequately support the workflows by technological means. Since ‘invisible work’ (cp. [63]) is often done by women, it is particularly their work that remains undervalued, since designers either ignore its importance for the organization as a whole or its complexity.

Design for women obviously risks celebrating stereotypes about ‘women’, their preferences, skills and work, which should rather be avoided. A de-gendering methodology, therefore, has to aim at attributing equal competencies to female and male users and upgrading women’s work. Hence, designers should strive to inscribe gender equality into technologies, if they are designed for female users, e.g. at women’s workplaces–as opposed to the case of technologies for general use, where they should become aware of gender differences and the diversity of users.[7] As already mentioned, in the latter category of technologies user tests seem to be a useful tool for recognizing that software and user interfaces do not fit the intended real user. In the example of the round dialog box it was demonstrated that, regardless of their gender, all test persons preferred a rather squared and axially symmetrical layout of the dialog box and strongly disliked the ‘female’ user interface (cp. [65]). Thus, the gender stereotype that women like curvilinear features, while men prefer squared ones, was clearly disproved.

However, if we want to move from analysis to an alternative design for the cases mentioned above, it is not enough to remain at the level of aiming to map social realities of work, life and use as best as we can, since such approaches tend to reproduce the existing structural-symbolic gender order. Hence, if technologies need to be designed for a predominantly female group of users it takes more than only applying user-centred design methods and evaluating usability. For a de-gendered design of such technologies, an explicit political positioning for those who are structurally discriminated seems necessary. The most well-known research to support workplace democracy and establish better working conditions for workers and employees through the use of technology is the Scandinavian tradition of participatory design (e.g. [17]). Following this approach, a variety of methods were developed and tested such as future workshops, design games and prototypes (cp. [31]). The aims and guidelines to ‘design for skill’ and ‘design for technical empowerment’ were already successfully applied in women’s workplaces such as nursing, office work or call-centre service work (cp. [16], [43], [56]). Since strategies against deskilling, degrading or learning to adapt and to program software in certain contexts work against the traditional gender hierarchy, these participatory design approaches can be considered as de-gendering methodologies if they are enhanced by a critical awareness of the gendered patterns in society and symbolism.

Representation of ‘the human’ in IT and the perpetuation of gender norms

A third category of gendered technologies include those that represent certain abilities, characteristics or even the nature of ‘the human’, but actually normalize gender stereotypical behaviour. Persuasive examples are human-like machines that explicitly display human bodies and human behaviour such as anthropomorphic sociable robots or emotional software agents. The bodily appearance of these artefacts, but also their concepts of action/behaviour and interaction/communication were exposed as intrinsically permeated by gender stereotypes (cp. [25], [71]).

Against a further consolidation of gender stereotypes, a de-gendering methodology should aim to de-construct the binary sex and gender system. This might be accomplished by artefacts that offer users and designers the possibility to gain an understanding of gender (and technology) as a social construction and instable, constantly performed and negotiated categories. A design philosophy that ‘allows users to engender themselves, to attribute to themselves a gendered identity of any one of a number of sorts, to create or perform themselves through using technology’ ([22], 204) is ‘underdetermined design’ (ibid.). While these ideas were primarily directed at encouraging gender identity formation in computer games for children that transcend gender stereotypes, ‘technology as experience’ [46] is an experimental account addressing ‘felt life’, which serves as a basis for broader design methodologies. ‘Design for experience’ [58], [59], as opposed to designing experience into an artefact, focuses not only on the subjective experiences (e.g. sensual, emotional, compositional, spatio-temporal) of the users, but also opens up space for potentiality and meaningfulness, i.e. a plurality of meaning construction processes that should not be closed or specified by design. ‘Reflective Design’ [60] goes one step further in stating that ‘reflection should be a core design outcome of HCI’ (ibid, 49). Reflection, in this case, is to be understood as critical reflection that renders users aware of unconscious aspects of experience. The methodology consists of principles and strategies, which combine the

[7] Here it becomes obvious that technologies reflect the well-known paradox of early feminist politics-aiming at equal opportunities (i.e. assuming gender equality), while at relying on—partly essentialist—distinctions between women and men, the female and the male realm (i.e. assuming gender differences). In order to resolve this paradox I suggest moving beyond an abstract category of technology. De-gendering strategies should rather be situated in the particular context of the artefact. A rough distinction might be whether the technology is designed for everyone (i.e. assumed neutral) or built for female (or male) users (i.e. assumed to address differences). Though this taxonomy does neither include—as we will see— ‘technologies of the self’ nor technological concepts in basic research.

analysis of the ways technology reflects and perpetuates unconscious cultural assumptions (such as the politics of race, gender and economy) with the design, building and evaluation of computational artefacts that reflect alternative possibilities. To my mind, this approach to provide technical support for self-reflection can be productively used to raise an awareness of gender stereotypes internalized by users, designers and artefacts.

Formalisms, abstract concepts and basic research: de-contextualization and objectivism

A fourth category of computational artefacts includes algorithms, formal objects and conceptual approaches in computer science that can be said to have gender politics. Abstraction, formalization and classification produce the impression of objectivity and a neutral research subject. Here, it seems problematic that these processes inevitable entangled with computer scientist's work disguise explicit as well as implicit decision made in the process of technology design, while they in effect establish hierarchies of knowledge, gendered classifications or dichotomies. Striking examples for this kind of gendering mechanism are algorithms and thresholds used to transform raw data from a computer tomograph into coloured pictures of the brain. It has been shown [38] that depending on the algorithm and threshold chosen gender differences of the brain appear—or do not appear. Thus, formalization is not innocent as also the field of knowledge representation and semantic web demonstrates. The knowledge, which is excluded from formal representation, has been revealed to correspond with the knowledge that is traditionally assigned to the female realm (cp. [1], [61], [9], see also [18]). Furthermore, dichotomies that underlie computational concepts might be symbolically gendered such as the dualisms of mind and body or rationality and emotionality. Attempts to overcome these dichotomies, e.g. in artificial intelligence, often consolidate a new, but also deeply gendered symbolic order (cp. [11]).

A de-gendering strategy for these formal objects presupposes a re-contextualization in use and in structural as well as cultural effects. It requires questioning assumptions, ontologies and epistemologies of technology design and a dissolution of dichotomies. Since such gendering processes are not necessarily referring to users and use, alternative technology design methods should mainly involve designers.

'Narrative Transformation' [67], [68] and 'Mind Scripting' [4] are approaches to identify presumptions and gender scripts in the technology design process. Both techniques are based on Frigga Haug's ‚memory work' that uses one-page long texts written by group members to deconstruct shared experiences and assumptions, which reflect societal structures. The authors transferred this idea from consciousness raising groups of the 1970ies to present groups of technology designers. 'Value Sensitive Design' [29], [30] is method that aims to inscribe certain desired values such as equity, diversity, inclusion into technological artefacts. It contains three levels of inquiry: empirical, technological and, most notably, conceptual studies that are based on moral philosophy and ethics. However, the method was already applied to computer game design by feminist scholars (cp. [27], [28]). In order to undermine or resolve dichotomies that are gendered in western traditional thought and connected to technology design, 'Critical Technical Practice' [2] can be a helpful tool. This method suggests to analyze designer's discourses, in order to identify key metaphors and then to invert these terms. It results in bringing in the margin to the centre of technology design. Another technique to question and change concepts of basic research in computer science are interventionist 'laboratory studies'. As a form of anthropological inquiry commonly used in science and technology studies (STS) laboratory studies are interventionist as such. However, applied in the context of basic research in computer science this method can explicitly be guided by feminist design goals such as the ontological and epistemological de-gendering of computational artefacts. This was successfully demonstrated by own research in the field of anthropomorphic software agents (cp. [12]). Hence, there already are a few techniques that seem promising to use for re-contextualizing formal objects and replace questionable ontological and epistemological assumptions in basic research in computer science.

RE-READING AND CRITICALLY EVALUATING THE METHODOLOICAL FRAMEWORK FROM FEMINIST THEORY

The approach roughly presented in this paper forms a basic methodological framework for a feminist technology design in computer science. It provides a broad spectrum of methods helpful for de-gendering computational artefacts. The suggestions are based on a thorough analysis of gendering processes. To make it a complete methodological framework, however, it needs re-reading from feminist theory and a practical evaluation. In this section theoretical traps and empirical improvement of the four strands are discussed.

When technologies are created "for everyone" designers often inscribe their own mental models into the technology. To avoid a gendering of the artefacts resulting from such an 'I-methodology' it was proposed to apply methods from "user-centered design". Such a de-gendering strategy is based on an equity argument and aims at the inclusion of users. In order to accomplish that, diversity and particularly gender differences have to be recognized. Thus, this strategy tends to essentialize alleged differences between women and man. Furthermore, user-centered design has the potential to adjust the technology to users by empirical means. If the users involved, however, only reflect traditional gender assumptions these methods cannot not bring critical or deconstructive impulses to technology design. Therefore, the choice of users that participate in the design process crucially affects the success of de-gendering attempts.

As a second gendering mechanism the inscription of gendered images of use, users and the division of labour into computational artefacts was identified. To support counteracting such gendering processes, which often occur in the development of software intended for female users, participatory design methods were suggested. Approaches in the Scandinavian tradition challenge existing societal structures of inequality and, thus, avoid reproducing the structural gender order by technological means, although they are sometimes built upon the rather simplistic assumption that emancipatory ideals could be inscribed into the technology. Applying these techniques from a feminist perspective often means to aim at making visible and revaluing women's work and competences. Hence, this strategy tends to re-essentialize gender. Moreover, making work visible is also ambivalent from political perspectives (cp. [18]). Thus, also the participatory design approach for de-gendering purposes needs a second reflection when applied.

Special technologies are representations of 'the human' in IT, which tend to normalize gender stereotypes. In order to deconstruct these gender assumptions (i.e. the existing binary sex and gender system) it was proposed to combine "reflective design" and "design for experience" with insights of current gender studies. Both approaches are based on a constructivist epistemology. From their theoretical bases that include the combination of technology design with critical social theory the method can easily connect with feminist theory and deconstructivist approaches. However, they have so far not been applied as de-gendering strategies. Furthermore, it is still an open question whether these methods can be applied to a broader scope of technologies of if there are more suitable techniques to deconstruct the binary sex and gender system by technology design.

Also, the methods proposed to avoid and counteract gendering of formal objects and basic research in computer science, which was caused by de-contextualization or disputable epistemological and ontological assumptions in the design process, need further empirical evaluation. Most of these were either developed for guiding software development and have to be transferred to basic research or they require testing if they can serve as a de-gendering strategy. Particularly in this area there is need of further research on de-gendering methods.

DE-GENDERING COMPUTATIONAL ARTEFACTS AS A FEMINIST TECHNOLOGY DESIGN METHODOLOGY

To sum up, this paper introduced a systematic approach to feminist technology design. It presented a starting point for a general methodology to de-gender computational artefacts and goes far beyond existing suggestions, which are either restricted to software applications or to problematic guidelines so far. The de-gendering approach proposed rather takes into account the complex gendering processes, which might occur in the field of computer science. Binding the approach back to feminist theory, however, warns us to use it as a simple recipe. It has been argued that analyzing the gendering and aiming at a de-gendering of computational artefacts needs a careful theoretical background, in order to avoid well-known shortcomings such as essentializing gender or technology or falling into the trap of technological determinism. Obviously, the approach needs empirical examination, further refinement and enhancement. Hence, the challenge is now to work with the proposed methodological framework to gain more experience with it, e.g. in a "de-gendering lab" located in a computer science research & development department.

REFERENCES

1. Adam, Alison. Artificial Knowing. Gender and the Thinking Machine, London: Routledge.1998.
2. Agre, Philip. Computation and Human Experience, Cambridge: Cambridge University Press 1997.
3. Akrich, Madeleine. The De-Scription of Technical Objects. In: Wiebe Bijker and John Law (eds.). Shaping Technology/ Building Society. Cambridge: MIT Press 1992, 205-224.
4. Allhutter, Doris and Hanappi-Egger, Edeltraud. The Hidden Social Dimensions of Technologically Centered Quality Standards. In: Dawson, Ray et al (eds.). Perspectives in Software Quality. The British Computer Society 2006, 179-195.
5. Archibald, Jaqueline, Judy Emms, Francis Grundy, Janet Payne, and Eva Turner (eds.). The Gender Politics of ICT. Middlesex: Middlesex University Press 2005.
6. Balka, Ellen and Richard Smith (eds.). Women, Work and Computerization, Boston: Kluwer 2000.
7. Barad, Karen. Posthumanist Performativity. In: Signs: Journal of Women in Culture and Society 28 (2003), 801-831.
8. Barad, Karen. Meeting the Universe Halfway. Durham & London: Duke University Press 2007.
9. Bath, Corinna. Interview http://www.semantic-web.at/1.36.resource.250.x22-all-animals-are-equal-x22-gender-research-a-fruitful-inspiration-for-building-semantic.htm [last access 8 February 2009].
10. Bath, Corinna. From Gender Analysis to Technology Design Methodologies. In: Arno Bammé et al. (eds.). Yearbook 2008 of the Institute for Advanced Studies on Science, Technology & Society. München/Wien: Profil 2009 (in print).
11. Bath, Corinna. Emotionskonzepte in der neueren Softwareagentenforschung. In: Mechthild Koreuber (ed.). Struktur und Geschlecht. Wiesbaden: VS-Verlag (in print).
12. Bath, Corinna. De-Gendering informatischer Artefakte. Grundlagen einer kritisch-feministischen Technikgestaltung. Dissertation am Fachbereich Mathematik und Informatik der Universität Bremen (in preparation).

13. Berg, Ann-Jorunne. A gendered socio-technical construction. The smart house. In: Judy Wajcman and Donald MacKenzie (eds.). The Social Shaping of Technolog., 2nd ed. Buckingham, UK/Philadelphia: Open University Press 1999, 301–313.

14. Bertelsen, Olav W.; Bouvin, Niels Olof; Krogh, Peter G. and Kyng, Morten (eds.). Proceedings of the 4th decennial conference on Critical Computing—Between Sense and Sensibility., New York: ACM 2005.

15. Beyer, Hugh and Karen Holtzblatt. Contextual Design. San Francisco: Morgan Kaufmann 1998.

16. Bjerknes, Gro and Tone Bratteteig. Florence in wonderland. In Bjerknes, Gro, Pelle Ehn and Morton Kyng (eds.). Computers and Democracy, Aldershot: Avebury 1987, 279-311.

17. Bjerknes, Gro and Tone Bratteteig. User participation and democracy. In: Scandinavian Journal of Information Systems 7(1) (1995), 73–98.

18. Bowker, Geoffrey and Star, Susan Leigh. Sorting Things Out. Cambridge, Mass.: MIT Press 2000.

19. Bührer, Susanne und Schraudner, Martina (eds.). Gender-Aspekte in der Forschung. Fraunhofer Institut für System- und Innovationsforschung. Karlsruhe 2006.

20. Butler, Judith. Gender trouble. New York: Routledge 1990.

21. Butler, Judith. Bodies that matter. New York: Routledge 1993.

22. Cassell, Justine. Genderizing HCI. In Jacko, Julie and Andrew Sears (eds.). Handbook of Human-Computer Interaction. Mahwah, NJ: Lawrence Erlbaum 2002, 402-411.

23. Crutzen, Cecile. Interactie, een wereld van verschillen. Een visie op informatica vanuit gender studies. Dissertation. Open Universitteit Nederland, Heerlen 2000.

24. Dix, Alan, Janet Finlay, Gregory D. Abowd, and Russell Beale. Human-Computer Interaction, New York: Prentice Hall 2000.

25. Draude, Claude. 'The agent that walked out of the display…"Unpublished paper. Berlin, September 2005.

26. Elovaara, Pirjo. Between Stability and Instability – Using ANT and ANTa as Analytical Perspectives Telling Information Technology Stories. In: International Journal of Feminist Technoscience (April 2007), http://feministtechnoscience.se/journal [last access 8 February 2009].

27. Flanagan, Mary; Nissenbaum, Helen; Diamond, Jim and Belman, Jonathan. A Method for Discovering Values in Digital Games. Full paper presented at Situated Play DiGRA '07 (Tokyo, JP September 24-28, 2007). Cited from: http://valuesatplay.org/?page_id=12 [last access 8 February 2009].

28. Flanagan, Mary; Howe, Daniel and Nissenbaum, Helen. Values in Design: Theory and Practice. In: van den Hoven, Jeroen and Weckert, John (eds.). Information Technology and Moral Philosophy. Cambridge: Cambridge University Press 2008. Cited from: http://valuesatplay.org/?page_id=12 [last access 8 February 2009].

29. Friedman, Batya and Kahn, Peter. Human Values, Ethics, and Design. In: Jacko, Julie A. and Sears, Andrew (eds.). The human-computer interaction handbook. Mahwah, NJ: Lawrence Erbaum Associates 2003, 1177-1199.

30. Friedman, Batya; Kahn, Peter and Borning, Alan. Value Sensitive Design. Technical Report 02-10-01, Dept. of Computer Science and Engineering, University of Washington, December 2002.

31. Greenbaum, Joan and Morton Kyng (eds.). Design at Work. Cooperative Design of Computer Systems, Hillsdale, NJ: Lawrence Erlbaum 1991.

32. Grundy, Frances; Köhler, Doris; Oechtering, Veronica and Petersen, Ulrike. Proceedings of Women, Work and Computerization. Springer 1997.

33. Hammel, Martina. Partizipative Softwareentwicklung und Geschlechterhierarchie. Frankfurt a.M.: Lang 2003.

34. Hansen, Wilfred. User engineering principles for interactive systems. In: American Federation of Information Processing Societies Conference Proceedings 39 (1971), 523–532.

35. Haraway, Donna. Simians, Cyborgs, and Women: The Reinvention of Nature. New York: Routledge 1991.

36. Haraway, Donna. A game of cats cradle. In: Configurations 1 (1994), 59–71.

37. Hofmann, Jeanette. Writers, texts and writing acts. In: Wajcman, Judy and Donald MacKenzie (eds.). The Social Shaping of Technology. 2nd ed., Buckingham, UK; Philadelphia: Open University Press 1999, 222–243.

38. Kaiser, Anelis, Kuenzli Esther and Nitsch, Cordula. Does sex / gender influence language processing? In: NeuroImage 22 (2004), Supl.1. Abstr. No MO39.

39. Kreutzner, Gabriele and Heidi Schelhowe (eds.). Agents of Change. Virtuality. Opladen: Leske + Budrich 2003.

40. Latour, Bruno. Nous n'avons jamais été modernes. Essai d'anthropologie symétrique. La Decouverte. Paris 1991.

41. Latour, Bruno. Reassembling the Social. New York: Oxford University Press 2005.

42. Law, John and Hassard, John (eds.). Actor-Network Theory and After. Oxford: Blackwell 1999.

43. Maaß, Susanne. Software-Ergonomie. Benutzer- und aufgabenorientierte Systemgestaltung', Informatik-Spektrum 16 (4) (1993), 191–205.

44. Maaß, Susanne and Els Rommes. Uncovering the invisible. In: Zorn, Isabel et al. (eds.). Gender Designs IT. Wiesbaden: VS-Verlag 2007, 97–108.

45. Marcus, Aaron. Human communication issues in advanced UIs'. In: Comm. ACM 36 (4) (1993), 100–109.

46. McCarthy, John and Peter Wright. Technology as Experience, Cambridge, MA: MIT Press 2004.

47. Nake, Frieder; Rolf, Arno and Siefkes, Dirk (eds.). Informatik zwischen Konstruktion und Verwertung. Bericht Nr. 01/04 des Fachbereichs Mathematik & Informatik der Universität Bremen 2004, 14-19.

48. Nielsen, Jakob. Usability Engineering, San Franscisco: Morgan Kauffmann 1994.

49. Oechtering, Veronika and Gabriele Winker (eds.). Computerplätze Frauennetze. Opladen: Leske + Budrich 1998.

50. Oost, Ellen van. Over 'vrouwelijke' en 'mannelijke' dingen. In: Margo Brouns, Mieke Verloo, Marianne Grünell (Eds.): Vrouwenstudien in de jaren negentig, ee kennismaking vanuit verschillende disciplines. Busson: Coutinho 1995, 287-312 (cited from Rommes 2002).

51. Oudshorn, Nelly. Genderscripts en technologie. Noodslot of uitdaging? In: Tijdschrift voor Vrouwenstudies 4 (1996), 350-367 (cited from Rommes 2002).

52. Oudshoorn, Nelly, Els Rommes, and Marcelle Stienstra. Configuring the User as Everybody. In: Science, Technology & Human Values 29 (1) (2004), 30–63.

53. Preece, Jennifer, Yvonne Rogers and Helen Sharp. Interaction Design: Beyond Human-Computer Interaction, New York: John Wiley & Sons 2002.

54. Rommes, Els. Gender Scripts and the Internet. Enschede: Twente University 2002.

55. Schavan, Annette. Der feine Unterschied. In: Die ZEIT, Nr. 17 (19 April 2007).

56. Schelhowe, Heidi, Maika Büschenfeldt, and Isabel Zorn. Das Sekretariat-Assistenz-Netzwerk (S-A-N) als Beispiel einer webgestützten Plattform für eine 'Community of Practice. In: Impulse aus der Forschung Universität Bremen 2 (2005), 10–13.

57. Schmitz, Sigrid and Britta Schinzel (eds.). Grenzgänge. Genderforschung in Informatik und Naturwissenschafte., Königstein/Taunus: Ulrike Helmer 2004.

58. Sengers, Phoebe. The engineering of experience. In: Blythe, Mark, Andrew Monk, Kees Overbeek, and Peter Wright (eds.). Funology: From Usability to Enjoyment. Dordrecht: Kluwer 2004, 19–30.

59. Sengers, Phoebe, Kirsten Boehner, Geri Ga, Joseph 'Jofish' Kaye, Michael Mateas, Bill Gaver, and Kristina Höök. Experience as interpretation. Paper presented at the CHI 2004 Workshop Cross-dressing and border crossing: exploring experience methods across disciplines 2004, http://www.sfu.ca/~rwakkary/chi2004_workshop/ [last access 8 February 2009].

60. Sengers, Phoebe, Kirsten Boehner, Shay David, and Joseph 'Jofish' Kaye. Reflective Design. In: Olav W. Bertelsen et al. (eds.). Proceedings of the 4th decennial conference on Critical Computing—Between Sense and Sensibility, New York: ACM 2005, 49–58.

61. Sherron, Catherine. Constructing Common Sense. In: Balka, Ellen and Smith, Richard (eds.). Proceedings of Women, Work and Computerization. Boston, Dordrecht, London 2000, 111-118.

62. Singer, Mona. Geteilte Wahrheit. Feministische Epistemologie, Wissenssoziologie und Cultural Studies. Löcker: Wien 2005.

63. Star, Susan Leigh. Invisible work and silenced dialogs in knowledge representation. In:Eriksson, Inger V., Barbara A. Kitchenham, and Kea G. Tijdens (eds.). Proceedings of Women, Work and Computerizatio. Amsterdam: North-Holland 1991, 81–92.

64. Suchman, Lucy. Human-Machine Reconfigurations. Cambridge: Cambridge University Press 2007.

65. Teasley, Barbee; Laura Leventhal, Brad Blumenthal, Keith Instone, and Daryl Stone. Cultural diversity in user interface design. In: SIGCHI Bulletin 26(1) (1994), 36–40.

66. Temm, Tatiana Butovitsch. If You Meet the Expectations of Women, You Exceed the Expectations of Men: In: Schiebinger, Londa (ed.). Gendered Innovations in Science and Engineering. Stanford: Stanford University Press 2008, 131-149.

67. Törpel, Bettina. Interest and Narration in Applied Information Technology - a strange combination? Second Tampere Conference on Narrative, Ideology and Myth, June 26-28, 2003.

68. Törpel, Bettina. Narrative Transformation: Designing Work Means by Telling Stories. In: Bertelsen, Olav W.; Korpela, Mikku and Mursu, Anja (eds.). ATIT - Proceedings of the first International Workshop on Activity Theory based practical methods for IT design. 2-3 September 2004, Copenhagen, Denmark. Århus, Denmark: DAIMI Technical Report #PB-574, 122-133.

69. Wagner, Ina. Women's voice. The case of nursing information systems. In: AI & Society, 7 (4) (1993), 295–310.

70. Weber, Jutta. Die Produktion des Unerwarteten. In: Corinna Bath et al. (eds.). Materialität denken. Studien zur technologischen Verkörperung. Transcript: Bielefeld 2005, 59-83.

71. Weber, Jutta and Corinna Bath. 'Social' robots & 'emotional' software agents. In: Archibald, Jaqueline et al. (eds.). The Gender Politics of ICT. Proceedings of

6th International Women into Computing Conference. Middlesex: Middlesex University 2005, 121–131.

72. Wiesner, Heike. Die Inszenierung der Geschlechter in den Naturwissenschaften. Frankfurt, New York: Campus 2002.

73. Winner, Langdon. Do Artefacts Have Politics? In: MacKenzie, Donald and Wajcman, Judy (Eds.). The Social Shaping of Technology. 2nd Edition. Buckingham, Philadelphia 1999, 28-40 (first published in 1980).

74. Zorn, Isabel; Susanne Maaß; Els Rommes; Carola Schirmer and Heidi Schelhowe (eds.). *Gender Designs IT*. Wiesbaden: VS Verlag für Sozialwissenschaften 2007.