Incommensurable Writings - Examining the Status of Gender Difference within HCI Coding Practices

Michael Heidt[1], Kalja Kanellopoulos[1], Arne Berger[2], and Paul Rosenthal[3]

[1] Chemnitz University of Technology, Research Training Group crossWorlds, Reichenhainer Straße 70A, Chemnitz 09126, Germany,
[2] Chemnitz University of Technology, Chair Media Informatics, Straße der Nationen 62, Chemnitz 09111, Germany
[3] Chemnitz University of Technology, Visual Computing Group, Straße der Nationen 62, Chemnitz 09111, Germany
{michael.heidt,arne.berger,paul.rosenthal}@informatik.tu-chemnitz.de,
kalja.kanellopoulos@phil.tu-chemnitz.de

Abstract. Gender relations are reproduced both within HCI development processes as well as within contexts of use. Hence, theorising the subject of gender becomes part of the responsibility of HCI as a form of practice. The fledgling subfield of feminist HCI has created an epistemological basis for thinking through these challenges. The current text seeks to relate to these contributions by analysing practices of coding as they pertain to HCI. We argue that coding is of yet undertheorised regarding the subject of gender relations. By drawing on the semiotic theories of Michael Mateas and combining them with Donna Haraway's reading of material-semiotic actors, the text aims to provide new impulses for a theorisation of the practice of code-writing as a (potentially gendered) writing practice. It thus aims at increasing the translatability of HCI theory into gender-aware communities of knowledge production.

Keywords: gender, coding, HCI, prototyping, interdisciplinarity, cultural informatics, critical technical practice

1   Introduction

Technological artefacts play a role in shaping our culture, while at the same time, cultural stances inscribe themselves within technological artefacts. During our daily lives, we constantly find ourselves exposed to environments populated by technological artefacts. This encompasses the context of culture, such as exhibitions, galleries, festivals and museums. As our lifeworlds are permeated by HCI artefacts, the scope of responsibility of HCI as a field widens. No longer, limited to the workplace, or to contexts such as 'entertainment', HCI has to deal with the issue of culture in all its breadth and complexity.

This implies that abstaining from issues of gender is no longer possible. These pervade culture; the artefacts we produce relate to them, whether we plan it this

way or not. Likewise, stances concerning gender are inscribed within the theories, processes and models used during our design activities.

There exists an extensive body of research addressing the practical [1], epistemological [2, 3] and methodological [4] implications of feminist HCI. However, the role of gender as it relates to the practice of code production has largely been overlooked. This might be due to the inapproachability of code, its lack of sensual, specifically lack of visual qualities.

Consequently, relating the fields of coding and gender discourses creates a conceptual challenge. Even more so, since many coders might not be inclined to discuss the relationship between these fields. Women might find a safe heaven within the community of coders where the incessant negotiation of gender seems to be suspended.

The present text argues that this is not due to an inherent gender-'neutrality' of code as an artefact, but constitutes a lack within contemporary theorisations of HCI development processes. We will subsequently outline a conceptual framework designed to relate the issues of code, code production and the performance of gender-relations.

2   Renegotiating Coding Practice

The discussion departs from an analysis of program code. Any coding individual within HCI finds herself confronted with a disparity. While her own gender is subject to ongoing negotiations, her texts are treated as technical and thus 'neutral' entities. In general, production of code is seen as a technical and thus gender-'neutral' activity.

Within 'Situated Knowledges' Haraway deconstructs claims for 'neutral' objectivity within scientific practice, instead highlighting the importance of 'writing technologies' [5, p. 595]. A theorisation of code that lends itself to an analysis in this fashion exists in the form of semiotic theories. Michael Mateas [6] develops a semiotic approach towards computing. He presents a view on systems that highlights coupling between rhetorical and technical strategies. Reintegrating this line of thinking with Haraway's reading of Latour [7], a new way of theorising code is facilitated: As coded texts enter into a network of cultural artefacts, they become part of processes in which gender relations are performed.

In the course of this analysis we will first discuss the notion of programming as a purely technical activity.

2.1   Culture / Material

Contrasting the view that programming deals with disembodied, purely logical entities is the view, that programs constitute material objects. This viewpoint is substantiated by historical reflection: Early computers were huge mechanical constructions, programs were specified in the form of physical objects, such as punched cards [8].

The view of software as something disembodied hinges on a clear distinction between hardware and software. However, this distinction itself has been critiqued numerous times.

Cramer calls for a suspension of the distinction between software and hardware [9]. In an approach that highlights the importance of imagination and phantasms he claims the history software to be exceeding that of conventional computers.

This line of thinking takes up themes that are apparent in Cixous' writing. If software creation is acknowledged as an expressive activity, one that allows for development of phantasms and imagined worlds its status might change.

Again, the present text warns of clear dichotomies. Assigning the imaginative, empathetic dimension of software production to the image of 'the sensitive' female can be read as just another strategy of further marginalisation. Instead acknowledging this phantasmatic realm allows for desires for the imagined worlds and the desire for pure instrumental reason to be redistributed between individuals. The division of labour between understanding, interpretive ethnography and calculating, technically productive code production might never have been convincing in the first place.

2.2 Programming Languages / Expression

Different programing languages, different programming styles are often conceptualised in analogy to tools. Any language makes solution of specific programs hard while facilitating solution of others. Different styles of programming are simply different problem solution strategies, equipped with specific advantages and shortcomings. However, if we follow writers such as Wardrip-Fruin calling for coding as *expressive* practice, the notion of style assumes a different role [10]. No longer conceptualised as a set of mechanical relationships, the text incorporates a translation of authorial traits. Following our earlier analysis, specification of a HCI artefact entails specification of a future relationship of author to a community of users. Thereby the relationship among users is remodulated.

Contrasting the activities of HCI-coding with practices of fiction writing might elucidate this relation. The story instructs its audience to live, to imagine an experience adhering according to the textual constraints laid down in writing. The interactive artefact provides a stage for its users, its digital components provide couplings on the level of mechanics as well as culture. Mechanical coupling is provided through the internal adhesion of formal systems. Cultural coupling points to the possibility that a program might be read by humans. Through symbolic names, comments as well as intelligible structure it conveys ideas to potential human readers. Returning to the example of the story, a program might be conceptualised more in analogy to an essay: A sequence of images, flowing into each other, connected by common strands of imagination. Thereby forming a loose fabric amenable to addition and unstitching. A small band of authors and positions seems to substantiate this lofty metaphorical view on the practice of programming [11–17]. At the same time, a program is legible by virtue of its

effects. Its actions within the shared world of humans and non-humans engender reconstructions and interpretations pointed at the opaque level of source code.

In effect, what is expressed is not only the workings of the machine but also the inscribed motivations, wishes of the community of writers. The language chosen acts as a translational device. The intention of a community, its hopes and phantasms are not able to copy themselves into a coded artefact. The way these are translated during the process of coding is partly determined by language. If one reads the code or experiences an artefact driven by it - a specific form of communication takes effect. In summary, appraisal of programming languages need not limit itself to their perceived ability of producing solutions. Their expressive faculties can be analysed, cherished or scorned.

2.3   Live Coding

The phenomenon of coding itself has entered the cultural realm, gaining relevance and visibility during recent years. As an example, the practice of *live-coding* stages actions of code production as cultural events [18–21]. The programmer is cast as a performer, her actions on the digital plane become directly readable, rendering her literary self transparent to the audience.

Again, coding itself is no longer pronounced a purely technical exercise. It is seen as part of the cultural activity of performative practice.

Explicit reflection of the practice within the discussed project context [22] proofed to be a stimulant for debate. Among other things it allowed for articulation of the question to what extent coding can be considered a site at which gender is performed.

At the same time, the often musical/auditive nature of live coding performance possibly opens the domain of coding to the rich sensual and imaginative realms pointed at by Cixous and other writers within the white-ink community.

Furthermore, the more direct mode of coupling between sensory organs and writing individual reconfigures the relationship between author and audience. Mechanical coupling between text and sound is complemented by perceptual coupling between sensing bodies within the audience. Conceptualising this mass as a single body possibly does not adequately describe the phenomenon. The achieved mode of coupling organises the audience into a collective body, a strange hybrid exhibiting multiple sexualities.

3   Case Study

The conceptual apparatus outlined was discussed, developed and refined in the course of a series of practice-led research studies. Three projects were conducted:

1. the development of a social-recommender system
2. development and deployment of an interactive installation aimed at instigating conversation among its users
3. integration of 1) + 2) into an integrated ambient information system to be deployed in museums

Analysed were digital as well as non-digital prototypes and mathematical formalisations alongside with their translations into source code.

During the study it became apparent, that although seldom addressed explicitly, the concept of gender pervaded discussions and development efforts in a tacit manner. Expediency of developed theoretical artefacts is described in reference to the case study.

3.1 Project Context

Division of Labour Within the project the dominating scheme for division of labour is that between social research and computer science. Computer science is usually framed as a technical enterprise, its ultimate goal being the production of digital technology. Social research on the other hand is responsible for interpreting and evaluating technological artefacts as well as for analysis of the domains and situations they are employed in. In effect this narrative entails a dichotomous rift. It speaks of a productive, yet unreflective and unimaginative sphere opposed to a reflective, analytical yet unproductive one.

Methodology Artefact development followed a methodology highlighting the productive role of difference between participants [23].
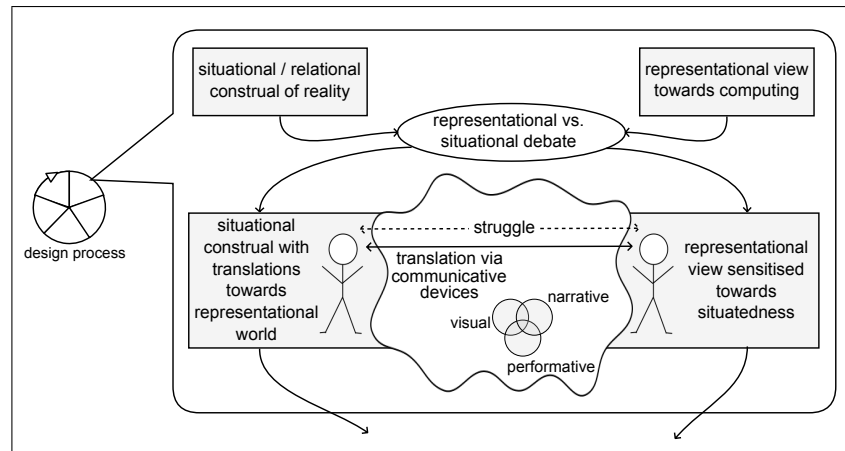

Fig. 1. Project Methodology, figure from [23]

Based on a critical reception of cross-epistemological social research [24] it focusses on mutual sensitisation as a means to render conflicts productive. Fundamental differences concerning interests and philosophical commitments [25, 26] are not resolved within discussion processes. Instead, the ensuing strife is used as a design resource in order to advance jointly conducted processes of collaborative making. As an example, the question whether reality is represented within

the scientific process or constructed and negotiated among researchers typically remains unresolvable between positivist and constructionist epistemologies. Debating the issue on this level will consequently not advance a making process. However, if a positivist researcher proposes inclusion of a timeline GUI element within a mobile application and a constructionist researcher proposes inclusion of a series of photographs forming a narrative arc, the debate becomes more productive. Both prototypes can be built on paper, it can be decided which course of action seems more promising within the context and limitations of the concrete situation of use. Hybrid forms become possible, combining elements of both prototypes without forcing researchers to adopt a corresponding 'hybrid epistemology'. Employed in this manner, diverging convictions become an asset. Participants developing different ideas and arguing for them passionately enrich making processes, while endless repetition of entrenched debate and following iterations of misunderstanding do not.

3.2 Interactive Installation

Developed was a full body interface that allows two participants to jointly experience historical narratives. Users enter the interaction space in pairs, marks on the floor guide their movements. Kinect sensors track the distance of each participant. While users move through space, biographical fragments are displayed on the projection thus forming a historical narrative. Users are free to explore history along this timeline, however a portion of the events reside within a special hidden critical zone which can only be explored jointly. In order to access the totality of content, users have to coordinate and move through the critical area in tandem. A more detailed description is available as part of an existing publication [27, 28]. Development proceeded by use of prototyping. Studio critique was used during early development iterations. Later stages relied on qualitative evaluation within 'real-world' deployments.

3.3 Mobile Recommender

The system initially was conceived as an interactive museum guide. As users move through the museum, they are presented with situationally appropriate information. Inspired by the Hippie-system [29], it was initially conceived as an information delivery device during early stages of development.

Its basic mode of operation consists of generating a recommendation involving two users: one that is able to explain the chosen item and another that is interested in the item.

Development was concept-driven, involving triangulation across different theoretical framings [31].

3.4 Observations

Code and Methodology An interesting lack was discovered pertaining to the level of code [32]. While qualitative research proved to be quite attentive to nearly
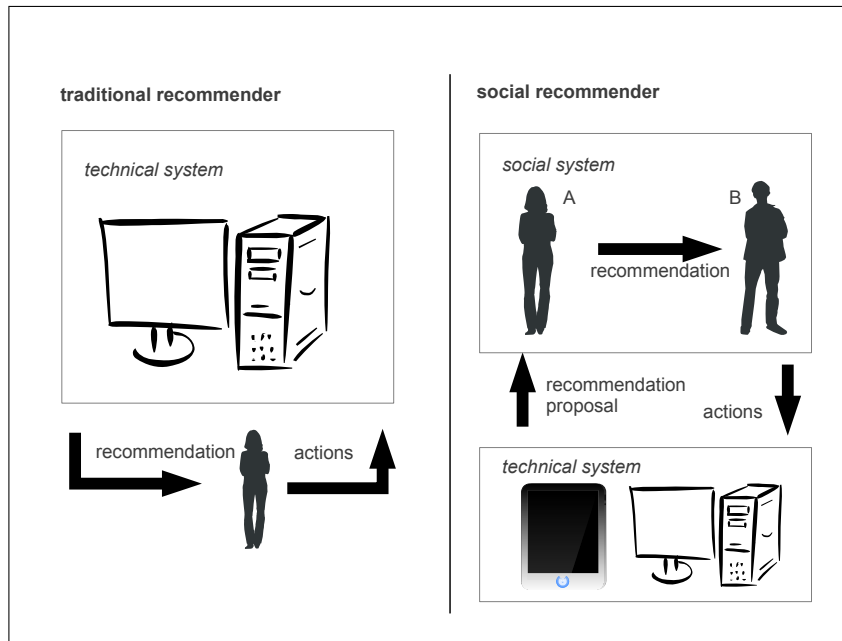
Fig. 2. Mobile Recommender - Figure initially appeared in [30]

every aspect of the artefact-making process, the level of code remained conspicuously absent. In effect, the division of labour remained remarkably strict. As a consequence, texts produced by coders were very seldom read by social researchers, while virtually every other aspect of their conduct remained relevant as a potential source of data. The stance subsequently adapted called for acknowledgement of coding as a practice related to processes of negotiation [33–35] as well as theory building [36].

Discussing Gender Gender issues were not discussed during phases of development and making.

However, during qualitative evaluation, observations were related to issues of gender. As an example, the (male) lead developer registered a higher preference of female participants to engage with the interactive installation. Factors such as the communal, non-confrontative aspect of the facilitated experience were subsequently discussed, in a manner that might be considered trite.

At the same time, the artefact itself and especially its digital algorithmic element was still perceived as a non-gendered object.

Performative Coding On an intellectual level, live coding practices were actively discussed within the project context [22]. These discussions led to a reframing of issues pertaining to user-generated-content (UGC). As a result, its limited applicability became apparent. Even in the face of UGC, the envisioned writing

process remained fraught with inequalities. UGC served as a specific assigned niche in which communal writing is allowed. More importantly, writing and the situation of interaction remain neatly divided.

While the potentials of live coding thus became apparent, none of the envisioned prototypes reached evaluation stages. However, the discussion succeeded to sensitise coding participants towards the topic of performativity. Topics such as the performance of gender relations [37] could subsequently be framed in a way more accessible to coding individuals.

4 Conclusion

Programming exhibits a double nature, being at once a combination of the imaginative and phantastical with the unwavering demands of formalised mechanisms. Analysing it this way can provide for an interesting answer to Haraway's challenge for providing a synthesis between radical contingency and a no-nonsense commitment to the 'real-world' [38, p. 187]. The program is indebted to its doubled audience: Adressing at any moment both the interobjective stratum of machinic actors and the intersubjective layer of human readers and interpreters.

The author thus becomes doubled as well. The space of coding offers a refuge from the injunction of gender in the form of a purely objective, technical realm consisting of formal relationships. At the same time, the human readership remains as ambiguous and divided as ever, forcing the author to relate in a way adhering to existing gendered power relations. The text cannot guard itself from being read in a gendered manner. Construed as a cultural artefact, it enters into the multitude of relations, repeating, among other things, sexual and gender identities. As far as coding or program execution is part of performative phenomena, it is to be read within the network of performative practices in which gendered relations reproduced themselves [37].

The concept of semiotic-material-networks provides novel impulses for theorising the status of code within HCI. By framing code production as a writing practice, existing theories and concepts can be appropriated in a fruitful manner. *Practice-led research practices* have proven to be an important methodological asset throughout the theory-making process. Reconceptualising the process of code production might render it more accessible to collaborators within the realm of social research. In highlighting to what extent it is a social practice, it could become a more intelligible phenomenon. Doing so would require both coders as well as social researchers to accept a renegotiation of the boundaries between both realms. In order to gauge the potential inherent within different styles of theorising, further empirical studies regarding the specificity of code writing within HCI seem to be warranted.

References

1. Bardzell, S.: Feminist HCI: Taking Stock and Outlining an Agenda for Design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '10, New York, NY, USA, ACM (2010) 1301–1310

2. Harrison, S., Sengers, P., Tatar, D.: Making epistemological trouble: Third-paradigm HCI as successor science. Interacting with Computers **23**(5) (September 2011) 385–392

3. Tatar, D., Harrison, S.: CHI 2011 Feminism , HCI and Control. In: Feminism and Interaction Design Workshop at CHI 2011. (2011) 1–7

4. Bardzell, S., Bardzell, J.: Towards a Feminist HCI Methodology: Social Science, Feminism, and HCI. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '11, New York, NY, USA, ACM (2011) 675–684

5. Haraway, D.: Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective. Feminist Studies **14**(3) (1988) 575

6. Mateas, M.: Semiotic considerations in an artificial intelligence-based art practice. Dichtung Digital: Journal on Digital Aesthetics (29) (2003)

7. Latour, B.: The pasteurization of France. Harvard University Press (1993)

8. Ceruzzi, P.E.: A history of modern computing. MIT press (2003)

9. Cramer, F.: Words made flesh. Code Culture Imagination. Piet Zwart Institute, Rotterdam (2005) 103–109

10. Wardrip-Fruin, N.: Expressive Processing: Digital fictions, computer games, and software studies. MIT press (2009)

11. Floyd, C.: Software Development as Reality Construction. In Floyd, C., Züllighoven, H., Budde, R., Keil-Slawik, R., eds.: Software Development and Reality Construction SE - 10. Springer Berlin Heidelberg (1992) 86–100

12. McLean, A.: Textility of Live Code. In: Torque#1. Mind, Language and Technology. Link Editions (September 2014) 141–144

13. Cox, G.: Speaking Code: Coding as aesthetic and political expression. MIT Press (2013)

14. Geiger, R.S.: Bots, bespoke, code and the materiality of software platforms. Information, Communication & Society **17**(3) (2014) 342–356

15. Rönkkö, K.: Interpretation, interaction and reality construction in software engineering: An explanatory model. Information and Software Technology **49**(6) (June 2007) 682–693

16. Fuller, M.: Elegance. Software Studies: A Lexicon (2008) 87–91

17. Krysa, J., Sedek, G.: Source code. software studies$\backslash$ a lexicon (2008) 236

18. Blackwell, A., McLean, A., Noble, J., Rohrhuber, J.: Collaboration and learning through live coding (Dagstuhl Seminar 13382). Dagstuhl Reports **3**(9) (2014) 130–168

19. Collins, N.: Live Coding of Consequence. Leonardo **44**(3) (May 2011) 207–211

20. Collins, N., McLEAN, A., Rohrhuber, J., Ward, A.: Live coding in laptop performance. Organised Sound **8**(03) (2003) 321–330

21. Dean, R.T.: Hyperimprovisation: computer-interactive sound improvisation. AR Editions, Inc. (2003)

22. Heidt, M.: Extemporising Digital Empowerment. In: Human-Computer Improvisation, Workshop at the ACM Conference on Designing Interactive Systems - DIS 2014, Vancouver, BC, Canada (2014)

23. Heidt, M., Kanellopoulos, K., Pfeiffer, L., Rosenthal, P.: Diverse Ecologies - Interdisciplinary Development for Cultural Education. In Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M., eds.: Human-Computer Interaction – INTERACT 2013. Number 8120 in Lecture Notes in Computer Science. Springer Berlin Heidelberg (January 2013) 539–546

24. Metcalfe, M.: Generalisation: Learning Across Epistemologies. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research **6**(1) (January 2005)

25. Blackwell, A., Wilson, L., Boulton, C., Knell, J.: Creating Value Across Boundaries; maximizing the return from interdisciplinary innovation. London: NESTA Research Report (2010)
26. Barry, A., Born, G., Weszkalnys, G.: Logics of interdisciplinarity. Economy and Society **37**(1) (February 2008) 20–49
27. Heidt, M., Pfeiffer, L., Berger, A., Rosenthal, P.: PRMD. In: Mensch & Computer 2014 - Workshopband, De Gruyter Oldenbourg (2014) 45–48
28. Wuttke, M., Heidt, M.: Beyond Presentation - Employing Proactive Intelligent Agents as Social Catalysts. In: Human-Computer Interaction, Proceedings HCI International 2014. Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014)
29. Oppermann, R., Specht, M., Jaceniak, I.: Hippie: A Nomadic Information System. In Gellersen, H.W., ed.: Handheld and Ubiquitous Computing. Number 1707 in Lecture Notes in Computer Science. Springer Berlin Heidelberg (January 1999) 330–333
30. Heidt, M.: Prototypengestütztes Reframing am Beispiel forschungsorientierter Systementwicklung. In: Mensch & Computer Workshopband. (2012) 191–196
31. Heidt, M.: Examining Interdisciplinary Prototyping in the Context of Cultural Communication. In Marcus, A., ed.: Design, User Experience, and Usability. Health, Learning, Playing, Cultural, and Cross-Cultural User Experience. Number 8013 in Lecture Notes in Computer Science. Springer Berlin Heidelberg (January 2013) 54–61
32. Heidt, M.: Reconstructing Coding Practice - Towards a Methodology for Source-Code. In Boll, S., Maaß, S., Malaka, R., eds.: Mensch & Computer 2013 - Workshopband, München, Oldenbourg Verlag (2013) 271–275
33. Curtis, B., Iscoe, N.: Modeling the Software Design Process. In Zunde, P., Hocking, D., eds.: Empirical Foundations of Information and Software Science V. Springer US (1990) 21–27
34. Strauss, A.L.: Negotiations: varieties, contexts, processes, and social order. Jossey-Bass (1978)
35. Strauss, A.L.: Continual Permutations of Action. AldineTransaction (2008)
36. Naur, P.: Programming as theory building. Microprocessing and Microprogramming **15**(5) (May 1985) 253–261
37. Butler, J.: Performative Acts and Gender Constitution: An Essay in Phenomenology and Feminist Theory. Theatre Journal **40**(4) (December 1988) 519–531
38. Haraway, D.J.: Simians, Cyborgs, and Women: The Reinvention of Nature. Routledge (May 2013)