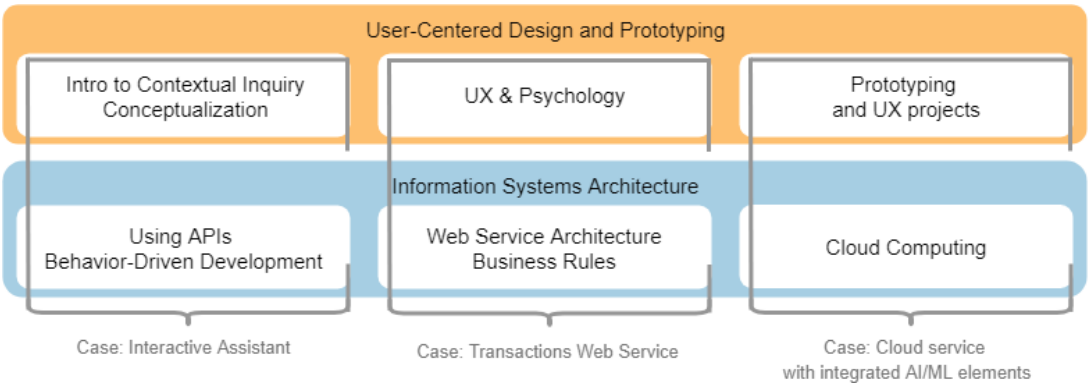
Co-aligning UX & Development Courses: The Case of MSc in Information Systems and HCI



Ilya Musabirov
National Research University
Higher School of Economics
St.Petersburg, Russia
ilya@musabirov.info

Alena Suvorova
National Research University
Higher School of Economics
St.Petersburg, Russia
asuvorova@hse.ru

Denis Bulygin
Uppsala University
Uppsala, Sweden
bulygindi@gmail.com

Pavel Okopnyi
University of Bergen
Bergen, Norway
pavel.okopnyi@uib.no

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored.

Table 1: IS&HCI courses

Year 1	
Intro to Databases and ER models	3 ECTS
Information System Architecture	8 ECTS
User-Centered Design	8 ECTS
Data Analysis	4 ECTS
Decision Making	4 ECTS
Cognitive Psychology and Social Behavior	6 ECTS
IT Project management	6 ECTS
Seminar 'HCI Research Methods'	6 ECTS
Projects	11 ECTS
Year 2	
Systems Analysis and Design	8 ECTS
Artificial Intelligence & Cognitive Systems	8 ECTS
User Interfaces	8 ECTS
NLP & Text Interfaces	4 ECTS
Customer Analytics (MOOC)	3 ECTS
Research Seminar	8 ECTS
Master's Thesis	14 ECTS

CCS CONCEPTS

• **Human-centered computing** → *Interaction design*; • **Social and professional topics** → *Computing education programs*; *Information systems education.*

KEYWORDS

user-centered design, information systems, curriculum

ACM Reference Format:
Ilya Musabirov, Alena Suvorova, Denis Bulygin, and Pavel Okopnyi. 2020. Co-aligning UX & Development Courses: The Case of MSc in Information Systems and HCI. In *Proceedings of EduCHI '20.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

INTRODUCTION

In this paper, we describe and reflect on our attempt to align the courses "User-Centered Design" and "Information Systems Architecture" to build a backbone for the newly created MSc program in Information Systems and Human–Computer Interaction (IS&HCI) at HSE University St.Petersburg. Table 1 outlines the program's curriculum.

While other MSc programs at the university's new School of Physics, Mathematics and Computer Science target students specializing in computer and/or data science, the MSc in IS&HCI is aimed at students with a computer science background who are interested in acquiring a deeper understanding of UX design principles in collaborative information systems, as well as social and behavioral science students who have some background in computer science or data analysis.

Creating a collaborative environment for such a diverse group presented a challenge in itself. Our goal was not only to encourage students with different backgrounds to work together, but also to transcend the patchwork nature of traditional course-based curricula, in which course boundaries define and restrict the scope of students' work. We also intended to give students experience participating in different stages of UX design and development from the very beginning of the program by enabling them to change roles in designer–developer interactions. Industry work often places recent graduates in preexisting polarized relationship dynamics between different participants in the design & development process. We believed that we could better equip students for work in their fields by providing them with early experience identifying trade-offs between design requirements

Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EduCHI '20, ,
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and architecture, as well as the opportunity to engage in supervised reflection on the process and a discussion of appropriate communication tools and practices.

As such, we focused on combining two main first-year courses, "User-Centered Design" and "Information Systems Architecture" along with integrating additional topics covered in the Research Seminars and Data Science classes. The main challenges we experienced are listed in the margins.

Challenges (Instructors Perspective):

- In practice, aligning two wide-ranging courses requires an investment in coordination between courses.
- Switching roles within student teams and developing based on the designs of others create pain points that require special care and inspire self-reflection.
- Students (whether or not they have a computer science background) tend to perceive design as more vague and thus less useful than coding and development.
- Students have a hard time distinguishing between similar concepts in different parts of the course (e.g., design vs. BDD scenarios, flavors of user stories).

COURSES DESIGN

The courses last three modules of a four-module academic year (eight months), which allowed us to create syllabi consisting of three iterations of design and development cycles (see teaser figure). These iterations are structured around three cases of increasing fidelity.

First cycle is dedicated to designing an interactive assistant and covers the basics of interaction design and contextual inquiry [8], keeping the discussion of interactions to a somewhat abstract level without much focus on UI. In this phase, the course also begins to discuss user stories as a "transfer agent" throughout the design & development cycle and helps students practice refining them using behavior-driven development approach scenarios as a way to specify acceptance criteria.

Second cycle is dedicated to redesigning architecture through different levels of prototyping for a transactional web service (e.g., curriculum self-planning on an MOOC platform or planning a vacation journey). This cycle builds on coursework in cognitive psychology. In the UCD component, the curriculum discusses concepts of usability, action models in HCI, and ways to support discoverability and feedback in user journeys [7, 14]. The last part of the module deals with interactive UI prototyping. In the architecture section, we discuss and practice creating multi-tiered web services with RESTful APIs.

Third cycle focuses on designing AI interaction experiences and expands on machine learning coursework. The UCD section discusses patterns and challenges related to designing AI-based services and focuses on evaluation techniques and data-related ethics. The architecture component of this cycle is dedicated to designing using cloud platforms, from the general purpose to the AI-specific.

This cycle is based on a data science course that lasts for the two first modules of Year 1. The course is HCI-oriented (for example, the statistics part of the course explains A/B testing), and covers traditional data analysis concepts, such as statistics, supervised and unsupervised machine learning, and issues of causal inference, as well as more specific topics. These topics include interpretable machine learning and explainable AI, which involves issues in machine learning models, both algorithmic (e.g., how to explore black-box models [1, 5, 6]) and social (e.g., bias [2], user trust [15], actionability [3], and explainability [12]). The detailed exploration of this topic has two purposes: first, it forces students to be more careful and skeptical when designing and analyzing AI-based services [9]; and second, it introduces them to the extensive research area of interpretability and explainable models and decisions in HCI [4].

The idea of combining AI and HCI is reconsidered on a more advanced level in the Year 2 course "Artificial Intelligence & Cognitive Systems," as well as the more technical course "System Analysis and Development of Complex IS."

STUDENTS' EVALUATION

After the first two modules, we conducted an anonymous survey which invited students to reflect on the joint project they completed for the two courses. The survey consisted of five open-ended questions about students' impressions of combining UCD and IS development courses in terms of topics, intersections, and the joint project. We asked students what they liked and disliked about this combination of courses, how they would change the program's structure in the future, and what skills they had already applied in side projects. Of the 18 students in the course group, we received 10 responses. Participants' answers are summarized below; quotes excerpted from students' feedback are provided in the margins.

"...you have the feeling of a real-life project where you explore the task from the different points of view. You have a whole toolbox to work with the project, not the separate instruments for each task."

The most commonly reported advantage of the joint project was its ability to enable students to understand concepts on a more general levelâĂŤnot only by applying new skills and acquiring knowledge, but also by implementing both development and design concepts in the same project. In this way, students learn how UX design and software development are interrelated and how concepts from each area interact with one another. Another important advantage for students is that they do not need to expend time and energy working on separate projects in each class, which allows them to focus on one particular topic.

"...it helped to consider the project in more detail to understand the relationship between application design and architecture."

One respondent also noted that the joint project improved their motivation to stay "on track." Since delays in one course automatically caused delays in the other, students were motivated to complete all coursework on time.

"[Delays during the classes]...that is, if we did not complete the model on UCD, we could not continue our work on Architecture."

On the other hand, the main reason to conduct one joint project for two courses is to demonstrate how two different areas of development interact. Sometimes it is necessary to discuss the same concept (e.g., job and user stories) twice, from both UX design and development perspectives. The overlap between these concepts can make students feel that they have been taught the same content repeatedly and that the learning process has stagnated.

"...at some points, the information was duplicated in both classes and there was a feeling that we were stuck in a time loop of job stories."

Moreover, some students felt that the interdependence between the two courses slowed the education process during the module. They described situations in which they felt that progress in one course was tied to progress in another. This interdependent content made them feel that, if they stopped working on a difficult topic in one course, they could not move forward in the other.

Another minor issue mentioned by students was working with the same team for a longer period of time, compared working with different teams on separate projects.

DISCUSSION AND REDESIGN IDEAS

While it presents various challenges, we view this attempt to align disciplines as important. Most of these problems and attitudes are not specific to this particular educational setting. While they can be masked by preserving and enforcing boundaries between different fields, we feel it is essential to continue working on the co-alignment, while taking into account students' and instructors' frustrations.

Our redesign approach builds on the Four Component Instructional Design (4C/ID) model of learning design [13]. The 4C/ID model shares our approach of centering complex experiences like those that would be encountered in the real world and unites cognitive approaches to learning with learning tasks, part-task practice, supportive information, and procedural information. In this spirit, we aimed to build a library of whole-task experiences, in which supportive information highlights different professional roles' perspectives on the case, facilitating both a holistic understanding of the task and an appreciation of possible demarcation lines between professional roles. Further task classes then shift towards the evaluation and discussion of possible interaction and communication issues related to these demarcation lines, as well as methods and instruments with which to bridge the gap.

Our study revealed that the "it works" experience of apps made programming a more rewarding activity for students. We are also working on finding an optimal balance between "pure" pen-and-paper, blank-sheet design experiences and the blueprints, templates, and instrumentation design characteristic of professional environments.

We believe that our challenges represent examples of more general issues, as summarized by Roudaut et al. [11], and that they can be addressed by exploring various ways of opening HCI curricula to integration, even in the form of single "knowledge bites" [10].

AUTHORS SHORT BIO

Ilya Musabirov is a Senior Lecturer in the Department of Informatics at HSE University St. Petersburg, Russia. He has professional experience as a data and systems analyst and information systems developer. He graduated with an MSc in informatics from ITMO University, Russia, and an MA in communication and media studies from Uppsala University, Sweden.

Pavel Okopnyi is a PhD student in information science at the University of Bergen, Norway. He has a professional background as a software engineer. He holds an MA in human–computer interaction from Uppsala University, Sweden, an MA in sociology from HSE University St. Petersburg, Russia, and a degree in engineering from Saint Petersburg Electrotechnical University ("LETI").

Denis Bulygin is a Lecturer in the Department of Informatics at HSE University St. Petersburg, Russia. He graduated from Uppsala University, Sweden, with an MA in human–computer interaction.

Alena Suvorova is an Associate Professor in the Department of Informatics at HSE University St. Petersburg, Russia, and the Academic Head of the university's MSc program in Information Systems & Human–Computer Interaction. She holds a PhD in theoretical computer science and specializes in computational decision-making models.

ACKNOWLEDGMENTS

We are grateful to all of the program students in the first and second cohorts, and Daniel Alexandrov whose role in creating the program was invaluable. We are also thankful to Frode Guribye and Barbara Wasson for all the advice, help and example.

The work is partially supported by Russian Science Foundation grant (project No. 19-71-00064).

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[2] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.

[3] Andres Colubri, Tom Silver, Terrence Fradet, Kalliroi Retzepi, Ben Fry, and Pardis Sabeti. 2016. Transforming clinical data into actionable prognosis models: machine-learning framework and field-deployable app to predict outcome of Ebola patients. *PLoS neglected tropical diseases* 10, 3 (2016), e0004549.

[4] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 80–89.

[5] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* 24, 1 (2015), 44–65.

[6] Christoph Molnar. 2019. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book

[7] Donald A. Norman. 2013. *The design of everyday things* (revised and expanded edition ed.). Basic Books, New York.

[8] Jenny Preece, Helen Sharp, and Yvonne Rogers. 2015. *Interaction Design: Beyond Human-Computer Interaction* (4 edition ed.). Wiley, Chichester.

[9] Mireia Ribera and Agata Lapedriza. 2019. Can we do better explanations? A proposal of user-centered explainable AI.. In *IUI Workshops*.

[10] Ariane Rodrigues, Dalai Ribeiro, and Simone Barbosa. 2018. Bringing Knowledge Bites to a PocketSize Curriculum for Teaching HCI. In *CHI 2018 workshop on Developing a Community of Practice to Support Global HCI Education.*

[11] Anne Roudaut, Orit Shaer, Audrey Girouard, and Andrew Kun. 2018. Identifying Challenges within HCI Education. In *CHI 2018 workshop on Developing a Community of Practice to Support Global HCI Education.*

[12] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296* (2017).

[13] Jeroen JG van Merrienboer and Paul A. Kirschner. 2018. *Ten steps to complex learning: A systematic approach to four-component instructional design.* Routledge.

[14] Stephen Wendel. 2013. *Designing for Behavior Change*. O'Reilly Media, Inc.
[15] Kun Yu, Shlomo Berkovsky, Dan Conway, Ronnie Taib, Jianlong Zhou, and Fang Chen. 2018. Do I trust a machine? Differences in user trust based on system performance. In *Human and Machine Learning*. Springer, 245–264.