Simulation-based Co-Creation of Algorithms

Sebastian Dennerlein[1], Dominik Kowald[1], Viktoria Pammer-Schindler[2], Elisabeth Lex[2] and Tobias Ley[3]

[1] Know-Center GmbH, Graz, AT
{sdennerlein,dkowald}@know-center.at
[2] Graz University of Technology, Institute of Interactive Technologies and Data Science, Graz, AT
{elisabeth.lex,viktoria.pammer}@tugraz.at
[3] Tallinn University, Institute of Informatics, Tallinn, EE
tley@tlu.ee

Abstract. Co-Creation methods for interactive computer systems design are now widely accepted as part of the methodological repertoire in any software development process. As the community is becoming more aware of the fact that software is driven by complex, artificially intelligent algorithms, the question arises what "Co-Creation of Algorithms" in the sense of end users explicitly shaping the parameters of algorithms could mean, and how it would work. Algorithms are not tangible like tool features and effects are harder to be explained or understood, especially in early design phases without a software prototype. Therefore, we propose a Simulation-based Co-Creation method that allows TEL researchers to collaboratively design algorithms with end users by creating user stories and personas, modelling assumptions and discussing simulated effects. The method extends the build & evaluate loop of co-design iterations, even when the learning technology for the algorithm is not ready. Our proposal is a methodological idea for discussion in the EC-TEL community, yet to be applied in a practice.

Keywords: Co-Design, Algorithms, Personas, User Stories, Simulation.

1 Introduction

The Co-Creation of interactive computer systems design has been addressed in the last decades. Methodological approaches such as Collaborative Design (Co-Design) and Design-based Research (DBRC, 2003) suggest the iterative creation of the design artefact with all stakeholders via successive prototyping. This also applies to the Co-Creation of Algorithms, but corresponding methods are missing. TEL researchers cannot leverage the same methods as for tools. For instance, algorithms can not be easily visualized in paper prototypes; their effects can only become partially visible in enacting a complex human-computer interaction with a paper prototype, which is cumbersome and not expressive enough. As algorithms are not tangible per se, their co-creation is more complex than co-creating learning tools. First, they can only be experienced when their interface to the user, i.e. the corresponding learning technology, is implemented and live. And due to the necessity of user interactions, second, they often need

time to show their effect in practice (e.g., facilitating consensus building in social tagging systems).

Designers of algorithms, however, need early and informed input from the target domain, so they do not lag behind the co-creation of the learning technology and do not follow wrong assumptions. Therefore, the open research question is: "How can algorithms be collaboratively designed for end users' needs so that the impact of the algorithm is easy to assess, especially in early phases of co-design without implementation in a live system?" We propose agent-based simulations for the co-creation of algorithms with end users to represent effects of algorithmic design decisions in a tangible manner before software prototyping. The proposed method extends the build & evaluate-loop of a co-design iteration by co-creating personas and user stories in each iteration that then serve as input for modelling the simulation.

2 Background and Related Work for Agent-Based Simulations

Agent-based simulations are computational models for simulating the interactions of autonomous agents and assessing their effects on each other and the whole system. One example of an agent-based simulation model is the Naming Game model (Baronchelli et al., 2006). Here, agents are represented as nodes in a network that interact at random. Nodes are equipped with so-called opinions and when two nodes interact, they either agree on a common opinion or extend their set of opinions if there is no common one. In the work of Hasani-Mavriqi et al. (2018), this model was used to easily simulate millions of interactions and study opinion dynamics and to assess the impact of various meeting rules (i.e., algorithmic parameters). Agent based simulations have also been used, for example, to study how recommendations are accepted by users (Saga et al., 2013). The latter is also used in the recently started European project CPN (https://www.projectcpn.eu/) to evaluate news recommendations.

3 Co-Designing an Algorithm using an Agent-Based Simulation

Co-Creation of Algorithms follows the known routines of a co-design iteration, i.e. the build & evaluate loop of stating assumptions in form of a (re-)designed prototype and evaluating it in the field (Ley et al., 2014). First, researchers need to establish an understanding of the domain. They can, for example, conduct a contextual inquiry to understand stakeholders, working practices and context, the algorithm needs to be co-designed for. This is reflected in user stories and personas for an abstract depiction of the needs and characteristics of the domain as well as their end users. Both kinds of design artefacts can be discussed and refined in each iteration. Personas are concrete, virtual people who are part of the target user group of the desired system and represent typical characters in abstract manner. They should not be thought of as a group as it is easier to think about concrete people than groups (e.g., Grudin & Pruitt, 2002). User stories (Thalmann & Schäper, 2018), on the other side, are abstractions of relevant practices including software systems, where a co-designed artefact should help to support or solve the design problem. They include the defined personas within their role,

reflect context as well as goal and depict interactions in a stepwise process of usage that can be visualized.

The acquired domain understanding then informs the (re-)design of the agents in the simulation as well as the algorithm itself: e.g. while the user story explains the circumstances and needs of the domain that inform the design of the algorithm, personas represent its stakeholders and serve modelling the agents for the simulation. Our proposal to use agent-based simulations, therefore, relies on the collaborative design process of creating and elaborating personas and user stories with end users in each co-design iteration. These design artefacts represent the core input to the iterative development of the agent-based simulation. As an example, let us assume the development of a novel recommender system. The user story suggests increased information sharing leading to the selection of two algorithms: Algorithm 1 recommends **popular** items and Algorithm 2 **similar** items as the ones the user has already interacted with. Let us further assume that we have two kinds of personas: User 1 consumes mainly **trustful** and agreed items and User 2 mainly **novel** items from her field of interest. These virtual users define the agents. Now, Algorithm 1 (**popular**) should perform better for User 1 (**trust**) and Algorithm 2 (**similar**) better for User 2 (**novel**), with both contributing to information sharing. With agent-based simulations, we can quickly assess such assumptions in early co-design phases without the need of users testing the recommender system live at that point in time. Furthermore, results of the simulation (evaluation) and a respective visualization can be fed back to and discussed with end users to adapt the algorithms (i.e., user stories) if they do not fit the virtual people (i.e., personas).

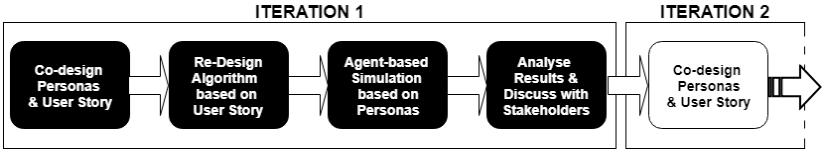

Fig. 1. Scheme of Iteration for Simulation-based Co-Design of Algorithms

The proposed process of co-creating algorithms with the help of agent-based simulations as part of a co-design iteration is reflected in Figure 1. We do not pose any other constraints on the co-design process, but simply assume in this paper that agent-based simulations may provide systematic input to the iterative co-creation of algorithm as part of (early) build & evaluate-loops. This way, a partly functional "prototype" can be implemented in form of a simulation representing a first draft of the algorithm. Based on such a simulation, researchers and users can assess the performance of the algorithm for the use case, even though there is no system available to implement the algorithm yet. As well, making the algorithm and its effects "observable" builds the ground for substantial discussions with end users allowing for revising the domain understanding in terms of personas and user-stories, for example. The higher understanding of the algorithmic impact may help (users) to find informed decisions, on the one hand, and may benefit the shared understanding of the co-design team, on the other hand.

We consider two further aspects of simulation-based co-creation of algorithms. In the early co-design phases, clearly the algorithm itself is a rough proposal, which must be wrong to a certain extent. This does not matter as insights are gained from wrong design decision as well. In later co-design phases, the algorithm might already be usable

as part of a running system allowing the end users to "directly" experience it. Analysing this real life-usage allows concretizing the domain understanding as well. Researcher can analyse the logs, compare the understanding of the users in the logs with the personas in the user stories as well as the agents in the simulation and update them. Afterwards, assumptions about the needs for the algorithm can be trialled with the end users by discussing the re-designed personas, user story and agent-based simulation. Hence, simulations could be drawn upon as security measure before risking live tests and potentially harming a running system; this becomes especially important when considering increasing levels of integrating the learning tool into (working) practice

4 Discussion and Outlook

We propose to co-create algorithms by designing user stories and personas with end users and respectively modelling agent-based simulations as part of each co-design iteration. A crucial role is the design of agents allowing for the early evaluation of the algorithm and getting usable insights before its implementation in the desired learning technology. We leverage the method not only for assessment, but understand it as a co-design method making algorithms tangible and researchable in a user- and domain-oriented approach, before a live system allows for that. This sheds light on the slightly "occult" process of algorithmic development and contributes to the much-needed conversation on how we ethically design these invisible decision-making instruments.

For future work, we plan to further concretize the proposed model and apply it for co-design of recommender algorithms in design-based research projects. It is also still to be clarified, if and how this methodological idea goes along designing and evaluating algorithms as part of a live system in later co-design phases

References

1. Baronchelli A, Dall'Asta L, Barrat A, Loreto V (2006). Topology-induced coarsening in language games. Phys Rev E.
2. DBRC, (Design-Based Research Collective). (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, *32*(1), 5–8.
3. Grudin, J., & Pruitt, J. (2002). Personas, Participatory Design and Product Development: An Infrastructure for Engagement. *PDC*, 144–152. https://doi.org/10.1.1.92.687
4. Hasani-Mavriqi, I., Kowald, D., Helic, D., & Lex, E. (2018). Consensus Dynamics in Online Collaboration Systems. Computational Social Networks Journal. SpringerOpen
5. Ley, T., Cook, J., Dennerlein, S., Kravcik, M., Kunzmann, C., Pata, K., … Trattner, C. (2014). Scaling informal learning at the workplace: A model and four designs from a large-scale design-based research effort. *British Journal of Educational Technology*, *45*(6), 1036–1048.
6. Saga, R., Okamoto, K., Tsuji, H., & Matsumoto, K. (2013). Evaluating recommender system using multiagent-based simulator: Case study of collaborative filtering simulation. In Lecture Notes in Electrical Engineering (Vol. 156 LNEE, pp. 155–162). https://doi.org/10.1007/978-3-642-28807-4_22
7. Thalmann, S., & Schäper, S. (2018). Localizing Knowledge in Networks of SMEs—Implication of Proximities on the IT Support. In K. North, R. Maier, & O. Haas (Eds.), *Knowledge Management in Digital Change* (pp. 189–206). Springer, Cham.