Software through the Perspective of Human-Centric Computing Paradigm

NexTech 2017 Keynote [Extended Abstract]

Evgeny Pyshkin
School of Computer Science and Engineering, Division of Information Systems, Software Engineering Lab
University of Aizu
Aizu-Wakamatsu, Japan
E-mail: pyshe@u-aizu.ac.jp

***Abstract*—This keynote focuses the aspects of software development and software education understood as human-centric computing applications. We discuss a number of our ongoing research and academic projects demonstrating how the integrated use of tools, techniques and methods referring to various disciplines can often contribute to a completely new technology domain and therefore can serve as the example of the transdisciplinary research.**

Keywords—human-centric computing; software; mobile applications; ubiquitos computing; computer-assisted learning; software development education.

I. Introduction

Human-centric systems and the systems based on human-computer interaction (HCI) technologies are substantially multidisciplinary [1].

On the one part, such systems naturally bridge the gaps between the disciplines involved in the design and implementation of computing systems supporting people's activities. On the other part, transdisciplinary research does not simply assume either the integrated use of tools, techniques and methods referring to various disciplines nor the interconnection among academic disciplines, but rather an application of solutions achieved in one area of knowledge (not necessarily "human-centric") to completely different application domains [2].

As far as 2003 in his famous "Leonardo's Laptop" book, Ben Shneiderman defined the shift from what computers perform to what users can do [3], the shift we continue to observe toady, in the age of digital transformation. The human-centric paradigm has flourished and transformed rapidly from an emerging area mostly limited to HCI issues to a wide domain with its own distinctive research agenda. It includes but not limited to designing solutions addressing human and societal problems and expectations, which are not just for or within a certain technology or context, as well as to developing personalized and adjustable interfaces for better and easier access to various information services.

HCC research efforts in advancing human-human and human-computer interaction are always interdisciplinary: they involve experts from different technology and social domains. In order to argue the above-mentioned considerations, we review a number of our ongoing projects as the examples illustrating the HCC interdisciplinary nature.

II. From Non-Native GUI and Image Recognition to Mobile Farms

One particular case presented in details in our regular NexTech-2017 conference series submission [4] is using image recognition algorithms for improving software testing automation process with a particular emphasis on applications with non-native or/and hand-drawn graphic user interface (GUI). As an example, we present the mobile game development project "World of Tennis: Roaring 20's" where we are involved in (http://worldoftennis.com/).

In this and similar systems a standard procedure of automated GUI testing (where we can access GUI controls programmatically nearly in the same way as users do) does not work, since all screen elements are in fact plain graphical images. Hence, constructing reliable test scripts require that we apply the models and algorithms of pattern recognition and information retrieval to the software testing process. Currently we are working on creating a framework aimed at providing a functionality of getting builds from a build machine, running all tests on all connected devices, generating reports with application action logs, screenshots and test execution statistics, as well as sending all the report-related data to the subscribed stakeholders. Since we require the tests that would run on real devices and need an extensible and feasible solution, we have to integrate all the parts into a distributed infrastructure that would allow users to use facilities of users' own computers and connected devices as the parts of the open testing infrastructure.

III. From Signal Processing to Language Learning

As noted in [5], computer-assisted learning is not limited to mere digitalization of learning process by transferring traditional techniques of managing language-related data with the use of computers. It should create totally new use cases that are hard to implement without computer technologies. Extending the concept of virtual labs to the case of language learning can be considered as one of such use cases [6]. In our ongoing project introduced in [7], we emphasize the aspect that high quality language learning is hardly accessible without attention to the particularities of non-verbal speech features, such as stress, rhythm, and intonation. Many language experts would agree that poor prosody could make speech unintelligible and incomprehensible. However, introducing the relevant

technology to a classroom is far from being a trivial problem: we have to address many practical issues of pitch processing including pitch detection, approximation and visualization with respect to modeling intonation styles and presenting them to a learner. Thus, the algorithms of signal processing meet problems of practical phonology and language learning. Introducing prosody-based models to a teaching process is promising for many languages, but especially useful for the languages where the stress is mainly phrasal (like in French) or tonic (like in Chinese).

IV. From Direct Application Interaction to a Companion Model

Developing components of information systems for travelers belongs to a particular subject domain of our research interests. In this domain, many current efforts are about delivering personalized and adjustable solutions following major scenarios we could learn from studying travelers' experience. In our recent work, we made an effort to address a problem of collecting information about user location within the context of traveling and investigate how the collected data are used. We realized that general-purpose apps (like the popular Google Timeline) often could not assure required location data quality properties (such as data collection frequency and location accuracy).

We introduced a model of mobile apps cooperation where an intelligent assistant (in our case, the application aimed at managing location data quality) tunes a general-purpose application by using data received through a service framework (e.g. Google Services Framework). Such an assistant allows travelers to continue using regular applications with achieving a desired level of location accuracy. In contrast to standard models (where the properties of existing applications are to be extended or improved), two applications do not interact directly between each other, but an existing one benefits from the fact that a companion is running.

V. From Engineering to Liberal Arts: Revisiting a Case of Software Engineering Education

Education is one of the ultimate human-centric technology applications. Similar to the methodology gap in software development in 1980-90's (i.e., the lag between software engineering theory and practical needs), now we probably observe a methodology gap in software education: there are many research efforts toward the analysis of software complexity, changeability, and difficulties of software design, modification and maintenance, however, only few inspired significant changes in teaching practices. Insights into examination of computer science curricula within the context of liberal arts give us a number of interesting considerations concerning developing students' soft skills, which are related to communication, collaboration and the abilities for public display of research projects [8].

Our experience to organize and teach a number of courses within the domain of software engineering allows us finding ways to transform one-way information transmission between a teacher and a student to their communication and, therefore, to better understanding of each other's perspectives, as well as to appreciating software developers' creativity requiring both engineering and humanitarian knowledge and abilities.

VI. Conclusion

The projects that we briefly described in this review are small examples of the tight integration of human sciences (e.g. social and cognitive) and computer science (e.g. HCI, signal processing, artificial intelligence, and ubiquitous computing). This integration gives us a hint to better understanding of the human-centric computing paradigm, which is about designing computer systems having, as nicely defined in [9], a human focus from beginning to end.

Acknowledgment

I would like to thank Prof. Maxim Mozgovoy and Prof. Vitaly Klyuev (University of Aizu), Dr. Vlatko Davidovski (Cognizant Business Consulting Strategy & Transformations), Prof. Natalia Bogach (Peter the Great St. Petersburg Polytechnic University), and Andrey Kuznetsov (T-Systems). Without the help of my respected colleagues, this NexTech 2017 keynote presentation would not be possible. I also thank Prof. Petre Dini (IARIA scientific program coordinator) for his valuable comments.

References

[1] H. R. Hartson, "Human–computer interaction: Interdisciplinary roots and trends," Journal of Systems and Software, Nov 30, 1998, vol. 43(2), pp. 103-118.

[2] E. Pyshkin, "Designing human-centric applications: Transdisciplinary connections with examples," In Proc. of 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), Exeter, UK, Jun 21-23, 2017, pp. 455-460.

[3] B. Shneiderman, "Leonardo's laptop: human needs and the new computing technologies," MIT Press, 2003.

[4] M. Mozgovoy and E. Pyshkin, "Using Image Recognition for Testing Hand-drawn Graphic User Interfaces," UBICOMM-2017, in press.

[5] E. Pyshkin, M. Mozgovoy, and V. Volkov, "Models and metamodels for computer-assisted natural language grammar learning," Social, Management, Economics and Business Engineering, vol. 9, no. 1, 2015, pp. 60-65.

[6] M. Mozgovoy and R. Efimov, "Wordbricks: a virtual language lab inspired by scratch environment and dependency grammars,", Human-centric Computing and Information Sciences, vol. 3, no. 1, 2013, pp. 1-9.

[7] Yu. Lezhenin, A. Lamtev, V. Dyachkov, K. Vylegzhanina, N. Bogach, E. Boitsova, "Study intonation: mobile environment for prosody teaching,". In Proc. of 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), Exeter, UK, Jun 21-23, 2017, pp. 448-454.

[8] E. Pyshkin, "Liberal arts in a digitally transformed world: Revisiting a case of software development education," CECR 2017, in press.

[9] A. Jaimes, N. Sebe, D. Gatica-Perez, "Human-centered computing: a multimedia perspective," In Proceedings of the 14th ACM international conference on Multimedia, Oct 23, 2006, pp. 855-864.