

## Panel Session CSQS: Conceptual Software and Quantum Software

**Chair:** Iaakov Exman

A high-quality panel should be thought provoking: to trigger questions, without any sort of definitive answers. The role of participants and of the Panel audience is to ask even more questions providing at most half-baked answers, leaving second thoughts for home. Thus, this Panel is restricted to a very short time duration.

It all started with a research proposal from MIT's Daniel Jackson, dated 2013 [1], from which we learned Fred Brooks' idea [2] that “*Conceptual Integrity* is the most important consideration for software system design”. It culminated with the recent thought provoking and excellent book by Daniel Jackson on “The Essence of Software – Why Concepts matter for Great Design” [3]. Yes, books should also be thought provoking. We wrote *culminated* very hesitantly, since it gives the false impression that this is the end of the story. As an antidote here are some trigger questions.

**Q1**<sup>1</sup> – Concepts matter for great design *or* Concepts are the very essence of Software?

**HBT1-** Focusing on apps' design implies that concepts are not the very essence of Software; Software is something else besides concepts. Our remark: “*yes and no*”. More about this, later on.

**Q2** – Is programming essential to Software? What is *the* programming language of Software?

**HBT2-** Emphatically *NO*. The natural language of software is the natural language of humans. Therefore, the Software vocabulary is constantly changing, together with novel technologies.

**Q3** – Is Software concepts' integrity enough to avoid the dire consequences of aviation or healthcare accidents? Particularly in cases where software has been proven to be faulty?

**HBT3-** Probably *NOT*. Any additional ideas?

**Q4** – Software Modularity should assure Conceptual Integrity? Algebraic representation of software, and its newborn child “Quantum Software Models”, manipulate vectors to be modular, without touching concepts at all. How is it possible?

**HBT4-** The Panel audience may decide to concisely discuss it, or may prefer to read papers [4],[5].

### REFERENCES

- [1] Daniel Jackson, “Conceptual Design of Software: A Research Agenda”, MIT-CSAIL-TR-2013-020 August 2013.
- [2] Fred Brooks, *The Design of Design*, Addison-Wesley, Boston, MA, 2010.
- [3] Daniel Jackson, *The Essence of Software*, Princeton University Press, Princeton, NJ, 2021.
- [4] Iaakov Exman and Alon T. Shmilovich, “Quantum Software Models: The Density Matrix for Classical and Quantum Software Systems Design”, 2021 IEEE/ACM 2<sup>nd</sup> Int. Workshop on Quantum Software Engineering (Q-SE), within ICSE'2021, <http://arxiv.org/abs/2103.13755>, 2021.
- [5] Iaakov Exman and Alexey Nechaev, “Quantum Software Models: Software Density Matrix is a Perfect Direct Sum of Module Matrices”, in Proc. SEKE'2022, pages 434-439, this Conference, 2022.

---

<sup>1</sup> **Q1** obviously means **Question 1**, while **HBT1** means **Half-Baked-Thoughts 1**.

## Conceptual Software - Panel Participants

There are three invited participants:

- Prof. Daniel Jackson – MIT – CSAIL – who is the Keynote speaker (short Bio under the Keynote information).
- Prof. Jonathan Aldrich – CMU – CS (short Bio below)
- Prof. Harold Thimbleby – Swansea University, UK – CS (short Bio below)

The Panel moderator will be

- Prof. Iakov Exman – JCE, Azrieli, Jerusalem, Israel – SE.

---

### About Prof. Jonathan Aldrich

Jonathan Aldrich is a Professor of Computer Science at Carnegie Mellon University. He teaches courses in programming languages, software engineering, object-oriented design, and program analysis for quality and security. Prof. Aldrich directed CMU's Software Engineering Ph.D. program from 2013-2019. Dr. Aldrich's research centers on programming languages and type systems that are deeply informed by software engineering considerations. His research contributions include modular and gradual verification of functional properties, typestate, and architectural structure, as well as the design of languages and type systems for usability. His notable awards include an NSF CAREER award (2006), the Dahl-Nygaard Junior Prize (2007), the DARPA Computer Science Study Group, and an ICSE most influential paper award (2012). He served as general chair (2015), program chair (2017), and steering committee chair (2017-2019) of SPLASH and OOPSLA. Aldrich holds a bachelor's degree in Computer Science from Caltech and a Ph.D. from the University of Washington.

### About Prof. Harold Thimbleby

Prof Harold Thimbleby is See Change Fellow in Digital Health, based at Swansea University, Wales. He is Expert Advisor on IT to the Royal College of Physicians, a member of the World Health Organization's Patient Safety Network, and an advisor to the Clinical Human Factors Group and to the UK Medicines Healthcare products Regulatory Agency (MHRA). Although a computer scientist, he is an Honorary Fellow of the Royal College of Physicians, the Edinburgh Royal College of Physicians, and of the Royal Society of Arts; he's also a fellow of the Royal Society of Medicine. He has been a Royal Society-Wolfson Research Merit Award Holder and a Leverhulme Trust Senior Research Fellow, and he is 28<sup>th</sup> Gresham Professor of Geometry (one of the oldest chairs of mathematics, with Henry Briggs - inventor of common logarithms - the first chair appointed in 1596). Harold won the British Computer Society's Wilkes Medal. He published *Fix IT: See and solve the problems of digital healthcare* (Oxford University Press, 2021). His previous book, *Press On: Principles of Interaction Programming* (MIT Press), won several international awards.