

Salon des Refusés: Dialectics for new computer science

Luke Church
University of Cambridge
luke@church.name

Tomas Petricek
Alan Turing Institute, UK
tomas@tomasp.net

1. MOTIVATION

Salon des Refusés (“exhibition of rejects”) was an 1863 exhibition of artworks rejected from the official Paris Salon. The jury of Paris Salon required near-photographic realism and classified works according to a strict genre hierarchy. Paintings by many, later famous, modernists such as Camille Pissarro and Édouard Manet were rejected and appeared in what became known as the Salon des Refusés¹.

A «Programming» workshop equivalent of Salon des Refusés can provide space for exploring new ideas and new ways of doing computer science. We welcome ideas that are difficult to evaluate and might even be seen as “unscientific”. The value of exploring such ideas is supported by significant evidence from history and philosophy of science (discussed throughout).

1.1 Workshop objective

The objective of the workshop is to provide a venue for discussing programming language ideas that are difficult to evaluate using established evaluation methods. The kinds of submissions we encourage and welcome are listed below.

THOUGHT EXPERIMENTS. Just as Galileo's early efforts involved thought experiments, analogies and illustrative metaphors rather than detailed experimentation², we believe that thought experiments can provide novel insights and inspire fruitful programming language ideas.

Wadler's widely cited (and never formally published) “expression problem”³ can be seen as such a programming language thought experiment. It defines a context for assessing abstraction capabilities of type systems, but it does so without requiring concrete definition of what a type is⁴.

EXPERIMENTATION. We find prejudices in favour of theory, as far back as there is institutionalized science⁵, but programming can often be seen more as experimentation than as theorizing. We encourage submissions that report stupendous discoveries⁶, even if there is yet no overarching theory that explains why they happened.

PARADIGMS. All scientific work is rooted in a scientific paradigm or scientific research programme⁷. Those define not only appropriate methods for answering scientific questions, but also frame what questions can be asked. For example, the Algol research programme seeks to increase confidence in correctness by the use of formal methods⁸.

We encourage submissions that explore alternative scientific paradigms or research programmes by acknowledging that logically perfect versions of theories usually arrive long after imperfect versions have enriched science⁹.

METAPHORS, MYTHS AND ANALOGIES. Any description of formal, mathematical, quantitative or even poetical nature still represents just an analogy¹⁰, and despite the dominance of mathematical and quantitative analogies, we believe that there are fruitful ideas that can be learned from other forms of analogy¹¹. After all, John von Neumann's First report on EDVAC¹², which introduced modern computer architecture, was inspired by a biological metaphor and referred to individual computer components as “organs”.

FROM JOKES TO SCIENCE FICTION. Ideas first presented as science fiction stories or said as a joke may enrich serious science in unexpected ways. The idea that a steam engine could be used to execute laborious computations was first suggested “in a manner which certainly at the time was not altogether serious” sparking “serious consideration of the possibility of mechanical computation.”¹³

A story or an artistic performance may explore ideas and spark conversations that provide crucial inspiration for development of new computer science thinking.

1.2 Audience: Academics and programmers alike

The workshop provides a venue where unorthodox programming ideas can be discussed and we aim to attract diverse and open-minded audience with a range of backgrounds.

- By accepting submissions that do not require established academic forms of evaluation, we create a venue that is welcoming to not just to novel ideas from the academic community, but also to practitioners.

¹ Based on http://en.wikipedia.org/wiki/Salon_des_Refuses (retrieved 21 Oct 2016). We apologise to art historians for any mis-interpretations.

² Chalmers (1999), p106

³ Wadler (1998)

⁴ An extended discussion can be found in Petricek (2015)

⁵ Hacking (1983), p150

⁶ The original citation appears in Hacking (1983), p151

⁷ Kuhn (1970) and Lakatos (1975), respectively

⁸ Priestley (2012), p257

⁹ Feyerabend (2010), p8

¹⁰ von Foerster (2013)

¹¹ Similar calls have been made e.g. in economics by Sedlacek (2011)

¹² von Neumann (1945)

¹³ Priestley (2011), p22

- We welcome work that presents alternative perspectives on programming including, e.g., treating spreadsheets as programming languages, live coded music and work that widens access to the affordances of programming.
- We would like to attract submissions and attendees who are interested not just in programming, but also in philosophy of programming and we encourage submissions that are reflections on academic programming research.

Opening the workshop to ideas that do not fit established PL conferences and evaluation methods contributes to the diversity of ideas presented at «Programming».

1.3 Relevance: Creating new open-minded venue

Similarly to the main «Programming» track, we welcome submissions covering a range of topics related to programming and we intend to contribute to the open-minded and innovative nature of the conference. Salon des Refusés complements the main track by explicitly seeking papers that do not present traditional evaluations. Such submissions can present interesting and valuable ideas that would not be accepted at the main conference track, which requires strong evidence or compelling arguments.

We provide a venue where novel work that expands our way of thinking about programming can be presented, provided that it sparks the interest of the PC or the attendees.

2 ORGANIZATION

We organised Salon des Refusés at «Programming» 2017. The workshop attracted 16 submissions (out of which 8 were accepted) and between 20 and 40 attendees during the day, making it the largest workshop at «Programming» 2017.

This proposal follows the same successful workshop format. We want to keep continuity by inviting some of the last year's PC members, but aim to expand the reach by drawing half of the new PC members from the wider PL community and from former workshop authors.

WORKSHOP ORGANIZERS.

- Luke Church (luke@church.name), University of Cambridge, UK. Luke studies how to improve the experience that people have when dealing with complex systems. He has experience as the organizing chair of several workshops (PPIG 2016, PX 2017) and as a PC member of numerous events (LIVE 2017, PX 2017, PPIG and Salon des Refusés 2017, etc.).
- Tomas Petricek (tomas@tomas.net), Alan Turing Institute, UK. Tomas served as the chair of Salon des Refusés 2017 and is active in the industrial programming community. He organized a number of industry events (including several conferences focusing on the F# language¹⁴ and virtual fsharpConf conference¹⁵).

PROGRAM COMMITTEE. The following former PC members are willing to join the PC of the workshop again:

- Dominic Orchard, University of Kent
- Felienne Hermans, TU Delft
- Antranig Basman, Raising the Floor – International
- Sam Aaron, University of Cambridge
- Stephen Kell, University of Cambridge

We intend to invite further PC members ranging from the wider PL community and former authors such as:

- Javier Burroni, University of Massachusetts
- Mariana Mărașoiu, University of Cambridge
- Jonathan Edwards, Formerly MIT and HARC
- Richard P. Gabriel, IBM and DreamSongs Ltd.
- James Noble, Victoria University of Wellington

WORKSHOP FORMAT.

- We expect to follow the deadlines recommended in the call for workshops (with a submission deadline at the end of January and author notification on February 17).
- We welcome short papers (3000 words) and long papers (9000 words) as well as screencasts or interactive essays.
- For accepted papers, we plan to have 20-minute talk by the author, followed by a 15-minute review talk and further discussion led by the primary reviewer of the paper. This would let us have up to 8 accepted talks in a day.
- We expect the attendance will be similar to last year (ranging between 20 and 40 attendees during the day).

EVALUATION PROCESS. Salon des Refusés gives a place to interesting novel programming ideas that are difficult to evaluate. The evaluation process focuses more on how well the presented work explores (or inspires exploration of) novel points in the programming language design space. We see this as a valuable contribution on its own.

In the program committee, we follow a process inspired by “identify a champion”¹⁶ where program committee members select papers they find interesting or worth discussing and expose them to constructive criticism. The committee members championing the accepted papers will be responsible for giving a second (critical) talk and also for writing critical review or commentary that will be published, together with the original work in the workshop proceedings. As an example, see Salon des Refusés 2017 paper Hermans, Aldewereld (2017) with a critical review Petricek (2017).

These means of engaging with papers follows the core of scientific practice in that peer review is the key part of the process, but it accepts the fact (see Polanyi¹⁷) that the value of scientific work cannot always be precisely articulated and often relies on personal commitment of an individual.

¹⁴ For example: <http://fsharpworks.com/paris/2014.html> and <http://fsharpworks.com/mvp-summit/2015.html>

¹⁵ <http://fsharpconf.com/>

¹⁶ Nierstrasz (2000)

¹⁷ Polanyi (1958)

REFERENCES

- Feyerabend, P. (2010). *Against method*. Verso (4th edition). ISBN 1844674428.
- von Foerster, H., Pörksen, B. (2013). *Pravda je výmysl lháře*, Pragma, 9788073494315 (translated from *Wahrheit ist die Erfindung eines Lügners: Gespräche für Skeptiker*).
- Hacking, I. (1983). *Representing and Intervening: Introductory Topics in the Philosophy of Natural Science*. Cambridge University Press. ISBN 0521282462.
- Chalmers, A. F. (1999). *What is this thing called science?* Open University Press.
- Church, L., Marasoiu, M. (2016). A fox not a hedgehog: What does PPIG know? In *Proceedings of PPIG 2016*.
- Church, L. (ed.), (2016). Pre-print of the 27th Annual Workshop of the Psychology of Programming Interest Group. Available at <http://www.ppig.org>.
- Hermans, F., Aldewereld, M. (2017). Programming is Writing is Programming. In post-proceedings of Salon des Refusés 2017. At: <https://refuses.github.io/papers/writing-is-programming-is-writing/paper>
- Kuhn, T. S. (1970). *The Structure of Scientific Revolutions*. The University of Chicago Press (2nd edition). ISBN 0226458040.
- Lakatos, I. (1975). Falsification and the Methodology of Scientific Research Programmes. In *Can Theories be Refuted? Essays on the Duhem-Quine Thesis* (ed. Harding, S. G.), pp205-259. ISBN 9789027706300.
- von Neumann, J. (1945). *First Draft of a Report on the EDVAC*. University of Pennsylvania.
- Nierstrasz, O. (2000). Identify the champion. *Pattern Languages of Program Design 4* (2000): 539-556.
- Petricek, T. (2015). Against a universal definition of 'type'. In *proceedings of Onward! Essays*, ACM
- Petricek, T. (2017). Review: Programming is Writing is Programming. Online post-proceedings of Salon des Refusés 2017: <https://refuses.github.io/papers/writing-is-programming-is-writing/review>
- Perry, R. B. (1955). *Realms of Value: a Critique of Human Civilization*.
- Priestley, M. (2011). *A science of operations: machines, logic and the invention of programming*. Springer Science & Business Media.
- Polanyi, M. (1958). *Personal knowledge: Towards a post-critical philosophy*. University of Chicago Press.
- Sedlacek, T. (2011). *Economics of good and evil: the quest for economic meaning from Gilgamesh to Wall Street*. Oxford University Press. ISBN 9780199767205.
- Wadler, P. (1998). The expression problem. Sent to the Java-genericity mailing list.

Call for Papers: Salon des Refusés – Dialectics for new computer science

Salon des Refusés (“exhibition of rejects”) was an 1863 exhibition of artworks rejected from the official Paris Salon. The jury of Paris Salon required near-photographic realism and classified works according to a strict genre hierarchy. Paintings by many, later famous, modernists such as Édouard Manet were rejected and appeared in what became known as the Salon des Refusés. This workshop is the programming language research equivalent of Salon des Refusés. We provide venue for exploring new ideas and new ways of doing computer science.

Many interesting ideas about programming might struggle to find space in the modern programming language research community, often because they are difficult to evaluate using established evaluation methods (be it proofs, measurements or controlled user studies). As a result, they are often seen as “unscientific”. Rather than requiring detailed evaluation, this workshop provides a venue where interesting and thought-provoking ideas can be exposed to critical evaluation. Submissions that provoke interesting discussion among the program committee members will be published together with an attributed review that presents an alternative position, develops additional context or summarizes discussion from the workshop. This means of engaging with papers enables explorations of novel programming ideas and new ways of doing computer science.

Topics of interest

The scope of the workshop is determined more by the format of submissions than by the specific area of programming language or computer science research that we are interested in. We welcome submissions in a format that makes it possible to think about programming in a new way, including, but not limited to:

- Thought experiments – we believe that thought experiments, analogies and illustrative metaphors can provide novel insights and inspire fruitful programming language ideas.
- Experimentation – we find prejudices in favour of theory, as far back as there is institutionalized science, but programming can often be seen more as experimentation than as theorizing. We welcome interesting experiments even if there is yet no overarching theory that explains why they happened.
- Paradigms – all scientific work is rooted in a scientific paradigm that frame what questions can be asked. We encourage submissions that reflect on existing paradigms or explore alternative scientific paradigms.
- Metaphors, myths and analogies – any description of formal, mathematical, quantitative or even poetical nature still represents just an analogy. We believe that fruitful ideas can be learned from less common forms of analogies as well as from the predominant, formal and mathematical ones.
- From jokes to science fiction – a story or an artistic performance may explore ideas and spark conversations that provide crucial inspiration for development of new computer science thinking.

Format and review

We welcome short papers (up to 3000 words) and long papers (up to 9000 words) as well as screencasts or interactive essays. We intend to publish accepted paper on the web, but any format is welcome for the submission (authors can use the <Programming> paper template).

Key dates

- Deadline for submissions: February 1st 2018
- Notification of authors: February 17th 2018
- Early registration deadline: March 6th 2018
- Workshop at <Programming> 2018: April 9th – 12th 2018

Program committee

Submissions that spark interesting discussion will be preferred over submissions with detailed evaluation. We asked the PC members to write about their interests in order to help authors find topics of positions that will be of interest to the PC.

DOMINIC ORCHARD (UNIVERSITY OF KENT). Dominic frequently works within the research paradigm of using mathematics and logic as tools for understanding programs and computation. He is fascinated by times when this activity feels like shoving a square peg in a round hole, presenting an opportunity to think outside, or against, the paradigm or seek better tools within it.

FELIENNE HERMANS (DELFT UNIVERSITY OF TECHNOLOGY). Felienne likes to think about what is and what is not programming. She especially loves to help people be better at programming, while they might not be actively looking to get better, because they do not self-identify as programmers. As such she has worked on code smells and refactoring for Excel and for Scratch, a programming language for children.

ANTRANIG BASMAN (RAISING THE FLOOR – INTERNATIONAL). Antranig wants to see work that widens the audience for software by considering the role it might take in healthy societies, based around artefacts that work for everyone. These days, he codes exclusively in JavaScript, the language of the proles - in whom our hope lies. He is interested in work which challenges the assumptions we use to carve up our domain into separated disciplines. He is excited by the possibility that we are still in the prehistory of our subject, and that the principles and practices we have adopted so far may be completely faulty.

STEPHEN KELL (UNIVERSITY OF CAMBRIDGE). Stephen thinks that programming, as we know it, has unacceptably high human cost, and that we cannot solve this problem by escalation. We need programming systems that help us not to write more code, but to write less, combine, downsize and simplify code. He is a system-builder, interested not only in designing and building such programming systems, but in evolving existing systems in this non-traditional direction.

SAM AARON (UNIVERSITY OF CAMBRIDGE). Sam is a live coder working directly at the intersections of art, education and programming language research. He is particularly interested in exploring the notion of liveness within languages enabling him to consider code as an interface for direct manipulation. He is the creator of Sonic Pi - a live coding music synthesiser currently gaining traction by both school teachers and musicians alike.

TOMAS PETRICEK (ALAN TURING INSTITUTE, LONDON). Tomas is interested in work that challenges how we think about programming. He is interested in novel programming models, theory and practice of functional programming, tools for data-driven storytelling and data science, but also philosophy of science applied to programming.

LUKE CHURCH (GOOGLE AND UNIVERSITY OF CAMBRIDGE). Luke Church is a researcher at Google and the University of Cambridge. He studies how to improve the experience that people have when dealing with complex systems. For example: programming languages, configuration systems or animal behaviour.