Technical Dimensions of Programming Systems: major revisions

We thank the reviewers for their continued attention and feedback. The major changes we have made are as follows.

Definition of "system". We worked out a sensible explicit definition of what a programming system is and included it as early as possible in the paper.

L/O/A typology. We intended the three-way classification in section 3 merely to provide some rhyme and reason as to where our example systems were coming from. We did not intend it as a rigid scheme where membership in a category means we are making some profound or bold claim. The L/O/A-type naming seems to have misled readers into anticipating the latter, so we have relaxed the naming to "language-based", "OS-like" and "application-focused". We did refer back to this scheme in one specific context, in section 5.2, noting that most OS-likes (Unix, Lisp and Smalltalk) cluster in a region to the right.

Dark & evaluation. Section 5 has been clarified as an evaluation of the framework by means of the two example usages. We've included a summary table for the dimensional analysis of Dark in section 5.1 and annotated the screenshot for convenience. We appreciate the reviewer's desire for examples of the specific interactions, but we thought it'd be pointless to write down something already explained in one of Dark's tutorial videos, so we cited the latter instead.

References to strengthen claims. Our statement about "cool/interesting" was entirely from experience, so we relaxed it to something we could more easily support with references. These can be seen in a footnote with our remarks linking them to the point we're trying to make. Elsewhere we have addressed this by deleting the claim (e.g. for "holistic view") or adding references, for example to the various example systems like Java and Jupyter. We're afraid we don't understand the request to put explicit examples in the section on venues for publishing programming systems research; we'd already listed UIST, VL/HCC, LIVE and PX. If this was about specific publications instead, we've now referred to some in the footnote on page 3.

Socio-Technical dimensions? The fact of the matter is that almost all of our 22 dimensions are technical, with the possible exception of Learnability and Sociability being socio-technical, and some remarks in Integrity vs. Openness. On this basis, we strongly disagree with calling the contribution "socio-technical" dimensions or similar, as this would give a false expectation to readers. We appeal to the reviewer to permit us to keep the name "technical"—we put a lot of effort into thinking of names early on, but "technical dimensions" was the best out of perhaps only two viable choices. If this is still unacceptable, we will fall back on the other name, "design dimensions", but if this is still rejected we would need some help determining an alternative.

On the new length of the paper. Following approval from the editors, this revision includes the full catalogue of dimensions in the main body of the paper. While we acknowledge this makes it very long, we do not intend reviewers or readers to read this section from start to finish. Instead, it is meant as a reference to be used as needed, for example to get further detail on the dimensions we reference later for the Dark programming system. This lets us present the material as we originally intended and addresses a reviewer concern that it was awkward to refer to appendix material in the section on Dark. In summary, we always saw the dimensions catalogue as belonging in the main body, but for review purposes it may still be treated like an appendix.