Emma Tosch · Research Statement

Computer programs automate more tasks than ever before: data-driven algorithmic decision-making can inform consequential real-world outcomes in disparate domains such as judicial sentencing, self-driving cars, and massively open online courses. Consequently, an increasing number of **procedures that we traditionally do not think of as computer programs** are now either encoded in software, or interact with software.

Adapting procedures from their manual versions to their digital counterparts can remove some known **errors or threats to validity**, while introducing novel errors or threats that lie exclusively at the intersection of a domain and its expression in software. Novel execution environments such as social media platforms, human computation systems, or even environments for autonomous software agents present new challenges for reasoning about **software correctness**. Fortunately, techniques from programming systems — especially from the fields of programming languages (PL) and software engineering (SE), such as **static analysis** and **code generation** — can be used outside their typical settings (e.g., type-checking or compiler), in the context of these atypical domains.

My work has used language design and software engineering techniques to aid in the prevention, diagnosis, and correction of statistical biases in data analysis:

Stastistal View		Programming Systems View	
Threat	Instrument	Technique	Toolname/Project
Selection Bias	Field Experiments	DSLs, static analysis, formal verification, code generation	PlanAlyzer (Tosch et al. (2019a, 2021); Clary et al. (2022))
Measurement Bias Construct Validity	Surveys	DSLs, static analysis, dynamic analysis, information flow	SurveyMan (Tosch and Berger (2014), **Distinguished Paper Award**)
Confounding	Simulation	software testing, dynamic analysis	ToyBox (Tosch et al. (2019b), Clary et al. (2018))

Program analysis is a *method* for combating threats to the internal validity of data collection instruments.

While much of my work in is in the *application* of existing tools, techniques, and technologies to novel domains, the requirements that tasks in these domains demand can lead to fundamental research in e.g., programming language design.

The long-term view of my work is to apply programming systems principles in service of developing tools that help **democratize data analysis**, **promote citizen science**, and **facilitate auditing and regulation of complex software systems**. These third of these goals of has recently led me to explore new research problems in **law and PL** — an emerging area within the PL community that leverages recent advances in not only language design and verification, but also machine learning and natural language processing. A powerful emerging technology from this field is the ability to automatically lift natural language into formal language, reducing the need for manual encoding and human annotation, which historically has been a barrier to formal verification of, e.g., legal statutes.

Experimental Design as a Programming Task

OOPSLA 2019, USENIX Security 2022; SIGPLAN Research Highlight 2020, CACM Research Highlight 2021; NSF FMitF-2220422 "FMitF Track I: Formal Methods in Software Support for Sound Experimentation", $661,021 (2022-2026)

Experimentation increasingly drives consequential decision-making of major actors in digital or online public spaces. Large firms have developed sophisticated experimentation management systems, including software frameworks and domain-specific languages (DSLs) for designing, writing, deploying, and analyzing experiments at scale. Unfortunately, this work happens in walled gardens, away from the public eye. Researchers seeking answers to questions about human behavior in such systems must either settle for observational data released by the firms that operate the infrastructure, acquire permission to run experiments at the firm itself, or run small-scale experiments on similar, independently operated infrastructure. There are critical limitations to each of these arrangements: some effects of interest may not be identifiable from observational data alone, running experiments at private for-profit firms presents a host of both logistical and ethical issues, and smaller-scale replicas of such infrastructure may not have external validity.

One major challenge that undercuts these issues is that such socio-technical systems were not designed to allow for experimentation. As new domains and platforms develop, the underlying infrastructure ought to support experimentation best practices, both for scientific endeavours and to aid in transparency for regulatory purposes. Unfortunately, most of the existing infrastructure work has happened within for-profit companies, which may not release reproducible artifacts or open-source software. There are few incentives for firms to enable the competition to better experiment. Most critically, however, there is very little public work available on enabling verifiably sound experimentation in socio-technical systems, and it is unclear whether any formal methods have been applied in this space. My work on PlanAlyzer — a static analyzer for programmatically-defined experiments that focused on combating selection bias — is the first of its kind in its attempt to address this space issues (Tosch et al., 2019a, 2021).

Ongoing and Future Work. I am currently advising two PhD students on the first non-industrial attempt to formalize the experimentation-analysis pipeline in socio-technical systems. This pipeline will tie together hypothesis registration, treatment allocation, and downstream statistical analysis, tightly coupling each phase of the pipeline in software. We intend to submit this work to PLDI 2023.

Unfortunately there are no publicly available corpora for experiments. Thus, to evaluate the validity of this formalism and spur research in this emerging area, we are building a corpus of experiments drawn from "found" experiments on the web, published papers, and new experiments directly encoded in our novel formalism; for novel experiments we will be working with a CS education researcher to explore the space of experiments that practitioners might want to encode in a learning management system. Corpus building and empirical analysis of current practice in software-mediated experiments is critical to the success of this work. We intend to submit the results of our corpus-building and empirical analyses to software engineering and data science conferences (e.g., ICSE, FSE, CODE, or KDD).

We will prove that the experiments written in a DSL that comports with our formalism have identifiable effects, automatically generating estimators for end-users that are consistent with the registered hypotheses. Building upon existing work in gradual typing and embedded languages, we will then show how our DSL can be integrated into existing codebases and languages, obviating the need for a standalone DSL, which will increase the usability a likelihood of adoption. In close consultation with a domain expert in computer science education, we will implement our methods in an example socio-technical system: an open-source learning management system, e.g. Moodle.

Dynamic Data Collection Instruments as Programs

All data-driven analysis and decision-making starts with data collection. Data collection tasks can be arranged on a continuum, based on the amount of control a researcher has over the data collection process. Observational studies and machine learning tasks typically use existing data sets whose data generating process and method for selection may not be known (e.g., exported application logging).

My research focuses on cases where the researcher has some level of control over the data collection process, using platforms such as Amazon's Mechanical Turk (AMT) and Facebook. **There are many potential sources of bias in the data collection process**, but two in particular lend themselves to software-based solutions: *measurement bias*, where the tools we use can have unintended effects on the data we collect, and *selection bias*, where some aspect of our collection apparatus causes the data to not be representative of the intended population. My PlanAlyzer work addresses the latter in experiments; my SurveyMan work addresses the former in surveys.

Measurement bias can be induced by tools that have unintended effects on variables of interest. Online surveys are a common tool for social science researchers to collect data. However, *question wording* and *question order* can bias results. I developed SurveyMan — a **DSL and runtime system for designing, deploying, and debugging web surveys** that can help prevent and diagnose these biases in the data collection process via randomized question selection and ordering, under user-defined constraints (Tosch and Berger, 2014). This work was informed by collaborations with researchers in Linguistics and Labor Studies. We designed the DSL as a **spreadsheet-based language** that would integrate with researchers' existing practices. Because our language included branching and random selection, execution was nondeterministic, leading us to employ both static and dynamic analyses to verify and monitor correctness. This work won first place at the **Student Research Competition** at PLDI 2014, and a **Best Paper Award** at OOPSLA 2014.

Ongoing and Future Work. I have been working with a student on applying **quantitative information flow** to **programmatically-defined adaptive surveys**. We are looking at the integration of measures of information content in text responses and their possible correlations with other questions: text response questions are often reported in full or part, but can leak information. He is currently looking at stylometric analyses and dialect classification, with obfuscation as a potential remedy to this leakage. We are additionally interested in studying: (1) the use of privacy budgets to inform participant payment and (2) the modeling of competing survey threat models (e.g., tired participants vs. bots) as an optimization problem. I am also interested in working on co-authorship of surveys with ML tools, including machine translators and GPT-3. Once I have an interested student, I intend to pursue a collaboration with an interested colleague at Microsoft Research on using surveys to study textual entailment.

Software Testing for Learned Software

SurveyMan and PlanAlyzer both operate over programs designed to collect data across human participants. However, there are data collection issues present in analyzing large software systems as well, especially when those systems interact with other software such as **autonomous agents**. This line of research presents unique challenges for data collection due potential mismatches between the data a researcher can record and the data used for analysis, as well as uncertainty over the underlying variability in the data generating process (Clary, Tosch, Foley, and Jensen, 2018).

In some domains, researchers have more control over the platform in which they do experiments. One such domain where platform design deserves a more principled look is in the evaluation and **explanation of deep RL** agents. My coauthors and I developed ToyBox, a suite of environments that simulate a subset of games from the Atari benchmark suite (Tosch et al., 2019b; Bellemare et al., 2013). While deep RL agents may seem to have nothing in common with surveys or field experiments, the challenges we face in evaluating these agents are actually quite similar to human computation: both involve **non-inspectable, and potentially non-interpretable, autonomous actors** making long-term decisions in complex environments.

ToyBox environments are **fully parameterized**, supporting **low-overhead intervention** on game state that can be applied mid-game, during training. ToyBox therefore combines the "found" nature of Atari games with the features necessary for intervention. ToyBox has already influenced, and is being used in, several machine learning research projects in my former group. Furthermore, an early ToyBox prototype I designed and developed has been used by our collaborators at Charles River Analytics (CRA). This prototype supported basic research on the generalization capacity and robustness of a then-state-of-the-art DQN RL agent (Wang et al., 2015; Witty et al., 2018). ToyBox also includes a **behavioral testing framework**; This framework would not be possible without platform support for intervention.

Ongoing Work. Tests in ToyBox can determine if an agent is learning a **generalized behavior**. Tests are both contextual and statistical: i.e., they can wait for a given condition to be met before execution, and they support replication over both single test conditions, as well as substitutable elements in the environment. My coauthors and I have used ToyBox to **validate claims made about deep RL agents**. For example, while agents do learn to hit the ball in Breakout at angles and brick configurations that could never have been seen during training, these same agents do not exhibit the higher-level strategy of "tunnelling" claimed in Mnih et al. (2015). Instead, agents appear to follow a fixed pattern of targeting certain bricks in a single column. This example provides **strong evidence of the need for causal evaluation** of deep RL agents.

My research has also taken steps toward **automated experiments** for **explanatory AI** (Tosch, 2020). This work relies on encoding ontologies of concepts, which has connections to the PL concepts of objects and types. I found that explaining behavior in **relational and time-varying environments** required developing an explanation typology (random, trivial, frame-violating, and causal) and posited that only one explanation type would be satisfactory to end-users: those that were causal. I built a prototype of this explanation system on top of the ToyBox framework and have been working to apply it to other simulation environments and to evaluate my hypotheses with human subjects. There is a direct line from early work I did on stopping conditions for evolutionary algorithms to this automated experimentation work Tosch and Spector (2012).

PL+Law

The literature on **causal inference** and the difference between *effects of causes* and *causes of effects*. The former describes experimentation, but the latter describes explanation and many of the examples are concerned with legal reasoning. Long-term, I would like to explore the connections between experimentation and law; short-term, I have been working with several students in the area of **PL+Law**. This is new interdisciplinary work: I have been inspired by Basu et al. (2019), which formalized the transfer of interest in property as a programming language. We are specifically looking at two possible domains: insurance contracts and arbitration agreements. We would like to use the formalization process as a means to generate scenarios that are entailed by these legal documents, and use those scenarios test end-user understanding of the implications of agreeing to these contracts. I have additionally been working with a recent UVM graduate (soon to be 1L at University of Wisconsin Law School) in the study of informed consent for participation in experiments vis a vis terms of service agreements and modal logic.

References

Shrutarshi Basu, Nate Foster, and James Grimmelmann. 2019. Property conveyances as a programming language. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, pages 128–142.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47:253–279.

Kaleigh Clary, **Emma Tosch**, John Foley, and David Jensen. 2018. Let's Play Again: Variability of Deep Reinforcement Learning Agents in Atari Environments. In *NeurIPS 2018 Workshop on Critiquing and Correcting Trends in Machine Learning*.

Kaleigh Clary, **Emma Tosch**, Jeremiah Onaolapo, and David Jensen. 2022. Stick It to The Man: Correcting for non-cooperative behavior of subjects in experiments on social networks. In *31st {USENIX} Security Symposium*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level Control through Deep Reinforcement Learning. *Nature*, 518:529 EP –.

Emma Tosch. 2020. *System Design for Digital Experimentation and Explanation Generation*. Ph.D. thesis, University of Massachusetts.

Emma Tosch, Eytan Bakshy, Emery D Berger, David D Jensen, and J Eliot B Moss. 2019a. PlanAlyzer: Assessing Threats to the Validity of Online Experiments. In *Proceedings of the ACM on Programming Languages*, OOPSLA, pages 182–212, New York, NY, USA. ACM. **CACM Research Highlight**.

Emma Tosch, Eytan Bakshy, Emery D Berger, David D Jensen, and J Eliot B Moss. 2021. Planalyzer: assessing threats to the validity of online experiments. *Communications of the ACM*, 64(9):108–116.

Emma Tosch and Emery D. Berger. 2014. SurveyMan: Programming and Automatically Debugging Surveys. In *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications (OOPSLA)*, pages 197–211, New York, NY, USA. ACM. **Best Paper Award**.

Emma Tosch, Kaleigh Clary, John Foley, and David Jensen. 2019b. Toybox: A Suite of Environments for Experimental Evaluation of Deep Reinforcement Learning. *arXiv preprint arXiv:1905.02825*.

Emma Tosch and Lee Spector. 2012. Achieving COSMOS: A metric for determining when to give up and when to reach for the stars. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 417–424. ACM.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. 2015. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on MachineLearning*, ICML.

Sam Witty, Jun Ki Lee, **Emma Tosch**, Akanksha Atrey, Michael Littman, and David Jensen. 2018. Measuring and Characterizing Generalization in Deep Reinforcement Learning. In *NeurIPS 2018 Workshop on Critiquing and Correcting Trends in Machine Learning*.