The Gamma

Anonymous Authors
Unseen University
Discworld
e-mail address

ABSTRACT
TBA

INTRODUCTION
some background

Example
put some screenshot here?

Our Approach and Contributions
We present blah blah. Our contributions are:

- Novel language design based on design principles

- Open source implementation on the web with cool use cases

- Critical evaluation

Long-term goal of making journalism better

RELATED WORK
Visual tools.

Programming tools. Notebooks

Journalism. Idyll

OVERVIEW
some walkthrough illustrating thegamma with code and screenshots

DESIGN PRINCIPLES
The design of TheGamma is based on two categories of design goals that we identify in this section. The first group of goals follow the observation that majority of users will not be expert programmers and can only invest limited amount of time into learning a new tool. The second group of goals are derived from our focus on journalistic applications and the specific challenges of this domain.

Principles for designing simple language
Our approach originates in a careful consideration of the challenges faced by users learning how to use main-stream programming languages for data science scripting. [**todo**: There must be some research on learning programming we can cite]

[**todo**: To me, some of the things below sound kind of trivial. Is that just me, or do we need more fundamental principles?]

A1. Allow learning from examples. Users of tools such as spreadsheets often learn by looking at existing problem solutions [**todo**: Advait's PPIG]. Our design should allow this by making it possible to inspect and retrace steps used while solving a problem in an existing application.

A2. Allow learning by experimentation. Another principle of spreadsheets that we want to keep is the ability to experiment and see results immediately. Our design should allow users to try invoking an operation or modifying a parameter and quickly see if this leads to the desired results.

A3. Choice over construction. To minimize the amount of information that users have to learn and remember, our system should work in a way that allows constructing programs by choosing from options that can reasonably appear in a current context, rather than requiring users to recall particular syntax or exact identifier name. [**todo**: recognition over recall?]

A4. Power for power users. Some users of the system may, over time, become advanced users and the system should support those. In other words, the upper bound on what can be achieved should be well above the most common use cases. [**todo**: I think I got this idea of "boundaries" on what is possible from some paper, but cannot recall which...]

A5. Minimal viable scenario. At the same time, the complex features that power users might need should not affect the most elementary uses of the system and should remain completely hidden until needed. In other words, the lower bound on what one needs to know to use the system for basic tasks should be as low as possible.

TODO: Evaluating systems paper

Principles addressing challenges in journalism

In addition to consideration based on our focus on non-expert users, we also identify a number of principles that are specific to our target domain. Following a careful consideration of current debates in journalism, we identify the following principles. [**todo**: we should find some references on trends in journalism to motivate these]

B1. Trust through transparency. The system should allow fact checking of the analyses to build trust. This means that viewers should be able to determine what is the source of analysed data and how has the data been transformed.

B2. Opening the process to readers. Journalists are increasingly opening the way they work to readers in order to build trust. The ability to view how an analysis has been constructed should not be limited just to experts (say, by running a Jupyter Notebook from GitHub), but should be available to all interested readers.

B3. Providing meaningful engagement mechanism. Finally, the system should provide a mechanism through which readers can engage in a meaningful discussion. For example, it should allow them modify parameters of a data visualization in order to show how, e.g. different choice of countries affects the result.

SYSTEMATIC DESCRIPTION

Language

Code should be visible to allow (A1)

auto-complete friendly language allows (A2)

language design addresses (A3, A4) by making easy things easy and hard things possible (if you know the right syntax)

Type providers

DESIGN PRINCIPLES EVALUATION

SYSTEM EVALUATION

Something measurable

Case studies

Scalability

DISCUSSION

Study limitations

exploratory in nature so we do not make any quantitative claims about effects

not comparing against other systems

Design principles

Design issues

modifying code