



# Rethinking User Interfaces for the Next-Generation of Workflow Tools

By Ryan Rodriguez

Undergraduate, Department of Electrical Engineering at UC Santa Cruz  
Supported by the Team for Research in Ubiquitous Secure Technology, UC Berkeley



## Introduction

“Scientists prefer to work in code, but today’s workflow tools have a graphical focus.”

The focus of this research will be on the development of an experimental user interface that aims to accelerate the prototyping of workflows by providing real-time visual feedback and code generation functionality. Scientists prefer to work in code, but today’s workflow tools have a graphical focus. It was hypothesized that an effective interface for composing workflows would be able to leverage the flexibility of coding while still providing real-time visual interactivity. Secondly the interface could accelerate analyses by generating the code necessary to execute workflows. The Tigres API for next-generation workflow production calls for precisely this type of interface.

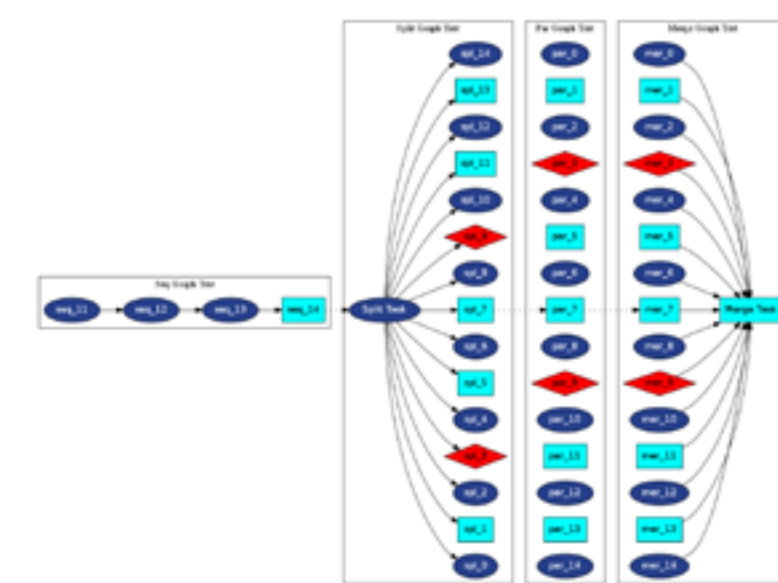
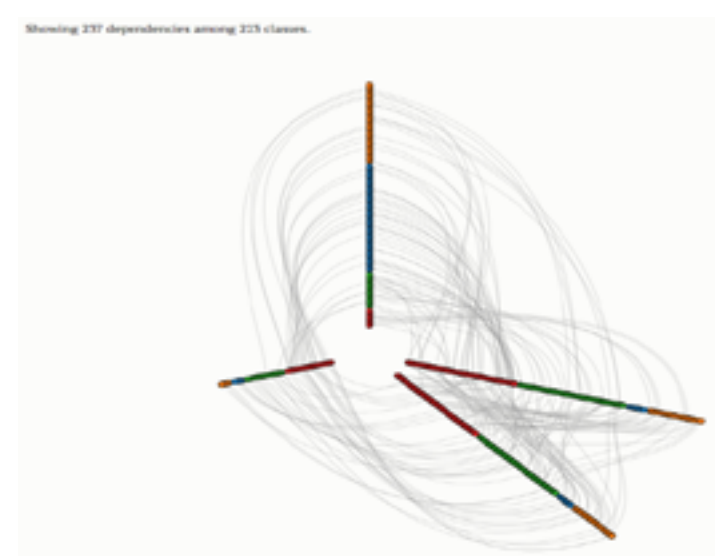


Fig. 1 A Workflow in Tigres

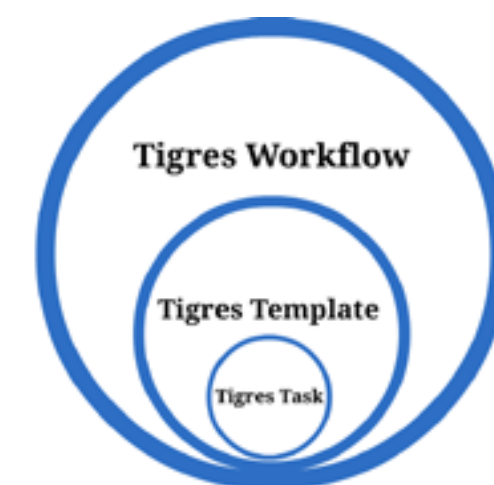
The Tigres project is a next generation scientific workflow API designed with ease of composition, scalability and light-weight execution in mind. The experimental interface known as the Active Console has been designed to meet these challenges. Inspired by Google’s real-time search bar and IDEs that allow for in-line execution, it’s intent is to allow for minimally intrusive execution and maximum flexibility. Using Python GUI capabilities alongside specialized visualization tools, it is possible to implement the three pillars of the interface known as the Active Console: the real-time visualization tool, the automatic code generation feature, and the console itself. The result is an interface that feels like an IDE, but provides the option of graphical interaction and compositional streamlining.”

“The result is an interface that feels like an IDE, but provides the option of graphical interaction and compositional streamlining”

## Background



- MapReduce is a popular programming model for parallel computing on a cluster.
- Implementing scientific workflows with a MapReduce implementation is difficult, even for experts.
- Workflow tools exist to address this problem, but their graphical interfaces sacrifice flexibility.

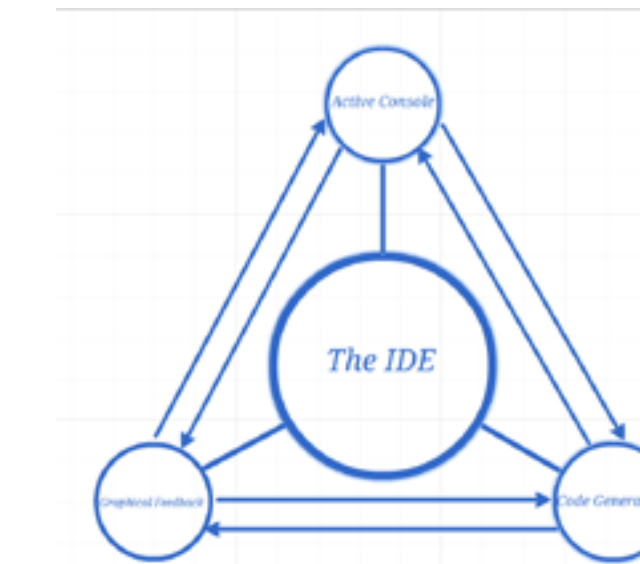


- Tigres was designed for composing and deploying scientific workflows to a diverse set of computational resources with ease
- It uses Templates that model common patterns in computing: Sequence, Parallel, Split and Merge. Each Template is assigned a set Tasks, which are functions or executables.
- The Tigres API has the potential to overcome the limitations of traditional workflow tools by leveraging the freedom of coding and the visual feedback of a GUI

## The Active Console

### The IDE:

- Users can compose workflows using the Tigres API with the IDE of their choice.
- The Active Console is executed alongside the IDE. It’s there when you need it, gone when you don’t.
- The Active Console interface consists of three components aside from the IDE: the console, code generator, and graphical feedback window.

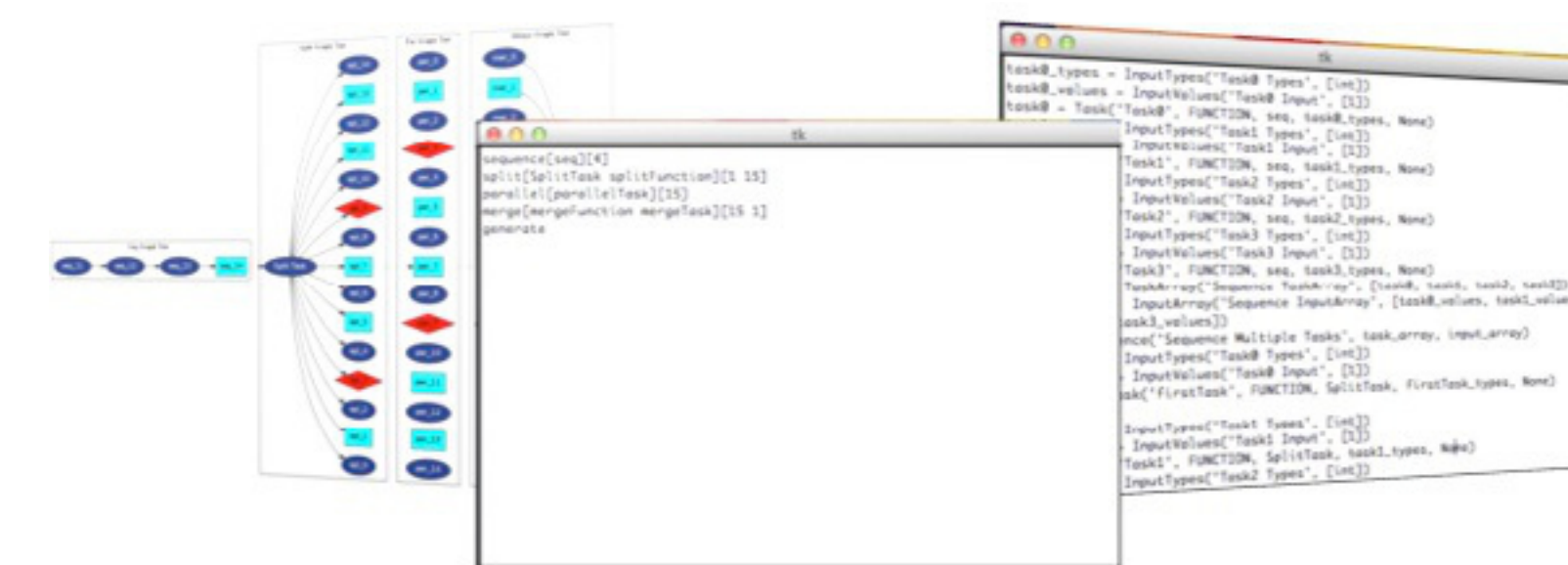


### Graphical Feedback:

- Graphs are generated in parallel with command entry. As soon as a user specifies a template in the console, it appears in the graph.

### Code Generation:

- Workflows are simple to design using the Tigres API, but can involve lots of typnig.
- With automatic code generation, menial coding tasks are reduced and productivity is accelerated.
- Code generation immediately reflects any changes made to the console.



### The Console:

- The console allows the user to specify large workflows rapidly using only six commands.
- The console is a tool that allows for bulk code generation using a reduced command set.
- By continuously parsing input, visuals and code are updated in real-time.

## Conclusions

- Through careful analysis of existing graphical workflow tools, the need for a more direct connection between code and graphic domains was discovered.
- By employing GUI elements that can be executed alongside existing workflows, the focus on the IDE remains paramount while the interface is available on demand.
- This paradigm leverages the flexibility and feature set available when coding in a modern IDE while preserving the visual capability of other workflow tools.
- Although graphical interactivity has remained somewhat elusive, the underlying principle of the Active Console has proved to be quite promising. The reduced command set and code generation features can dramatically reduce time from concept to workflow execution. Unlike the current generation of tools that draw a project into a box, the Tigres API with the Active console will give the user ultimate freedom through complete control.

## Acknowledgements

PI: Deb Agerwal

Research Scientist, Mentor: Lavanya Ramakrishnan

Core Team: Dan Gunter, Gilberto Pastorello, Valerie Hendrix



Contact Information:  
ryarodri@ucsc.edu

