

# Visual Programming Language Projects

Prof Ken Hawick

[k.a.hawick@massey.ac.nz](mailto:k.a.hawick@massey.ac.nz), [www.massey.ac.nz/~kahawick/student-projects.html](http://www.massey.ac.nz/~kahawick/student-projects.html)

A **Visual Programming interface makes use of pictorial icons and graphics** instead of raw programming text. Visual Programming systems can be based on an underpinning conventional programming language or can be entirely Visual Programming Language (VPL). There are several proprietary and public domain “builder environments” available. These are not full visual programming languages but are just an integration of graphical interface technology with programming editors used with conventional text based programming languages.

This project is to **develop prototypes of a graph-structured visual programming framework** for data-flow applications. Particular novel features for investigation include a data flow type system; hierarchical structuring and grouping of language components; and development of distributed computing infrastructure for distributed processing of language components.

Various **implementation technologies** can be considered including **Java; C/C++/OpenGL; Tcl/Tk**. A Java based system may be able to make good use of the JavaBeans concepts. It may be necessary to develop language and macro manipulation systems including partial scripting systems to support development of the main VPL system.

A first stage is to utilise a **graph object to hold the task graph** specified by the visual programme. This can use a **flat type-system** based on a small set of conventional primitives. A second stage involves **various graphical interface developments** to support user specification of the programming task-graph. A refining stage is to develop practical implementations for **managing hierarchical task-graphs**. More advanced stages involve: experiments with a **more elaborate and structured type system** for general programming; designing a support system for broad **development of applications-specific modules**; developing high performance and distributed type infrastructure to **make use of cluster and supercomputing resources**.

Parts of this project could be carried out at **honours, masters or PhD level** with obviously more work required to implement more of the aspects of the system at higher levels of study. This project will be able to build on graphics structure, type theory systems and other theoretical and practical apparatus already developed. This project will suit students with interests in programming languages and algorithms in general as well as graphical interface technologies. This project **spans many important technical areas in software engineering**.

