Structured graphs

a visual formalism for scalable graph based tools
& its application to software structured analysis

Submitted by Mark J. Sifer to meet the requirements of the Doctor of Philosophy in the School of Computing Sciences at the University of Technology, Sydney in 1996.

CERTIFICATE

I certify that this thesis has not already been submitted for any degree and is not being submitted as part canditure for any other degree.

I also certify that the thesis has been written by me and that any help that I have received in preparing this thesis, and all sources used, have been acknowledged in this thesis.

Signature of Candidate

Production Note:
Signature removed prior to publication.

Acknowledgement

I am indebted to my principal supervisor John Potter for both his guidance and collaboration in this work. I also wish to thank my co-supervisor John Leaney for his support. Some of the concepts in the second half of this thesis had their genesis while I was a research student at Hiroshima University, under the supervision of Tadao Ichikawa.

This work has also benefited from discussions and collaborations with Peter Eades, Masahito Hirakawa, David Lowe, and former colleagues in the Computer Systems Engineering Division of Computer Sciences Corporation (Australia). I also wish to thank the thesis examiners for their detailed comments. Addressing these, has significantly improved the clarity and presentation of the thesis. The work was funded by an Australian government postgraduate award with a top-up from the School of Computing Sciences.

Abstract

Very large graphs are difficult for a person to browse and edit on a computer screen. This thesis introduces a visual formalism, *structured graphs*, which supports the scalable browsing and editing of very large graphs. This approach is relevant to a given application when it incorporates a large graph which is composed of named nodes and links, and abstraction hierarchies which can be defined on these nodes and links.

A typical browsing operation is the selection of an arbitrary group of nodes and the display of the network of nodes and links for these nodes. Typical editing operations is: adding a new link between two nodes, adding a new node into the node hierarchy, and moving sub-graphs to a new position in the node hierarchy. These operations are scalable when the number of user steps involved remains constant regardless of how large the graph is. This thesis shows that with structured graphs, these operations typically take one user step.

We demonstrate the utility of structured graph formalism in an application setting. Computer aided software engineering tools, and in particular, structured analysis tools, are the chosen application area for this thesis, as they are graph based, and existing tools, though adequate for medium size systems, lack scalability.

In this thesis examples of an improved design for a structured analysis tool, based on structured graphs, is given. These improvements include scalable browsing and editing operations to support an individual software analyst, and component composition operations to support the construction of large models by a group of software analysts.

Finally, we include proofs of key properties and descriptions of two text based implementations.

Contents

Part II Structured Graph Components