



The Australian Australian Intelligent Tools Program

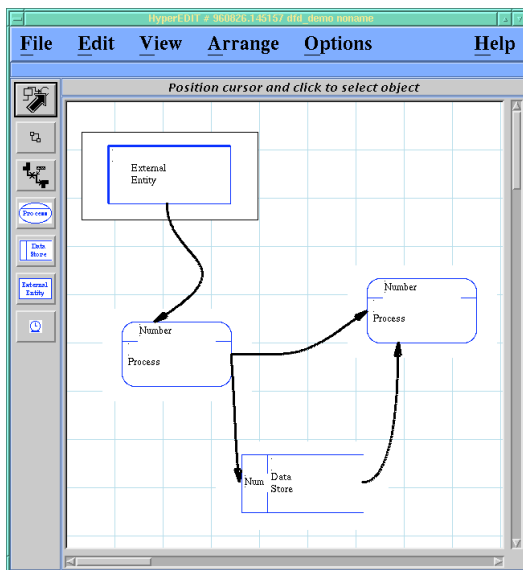
University and Industry Collaboration in Practice

The Amdahl Australian Intelligent Tools Program

University and Industry Collaboration in Practice

The Amdahl Australian Intelligent Tools Program is a medium scale research program hosted by La Trobe University and funded very generously by Amdahl Australia. The Program, part of Amdahl's participation in the Australian Partnership for Development Program, supports the HyperCASE Software Engineering project, and provides teaching and equipment resources to La Trobe University.

HyperCASE is an integrated system of software tools providing support for "front-end" software development, and which are integrated under a Hypertext system. Each of the tools is "Hypertext Aware", allowing documents and tools to be linked in a classic Hypertext paradigm. In practice, Hypertext transfers can be made between documents which are in fact displayed by separate tools running under independent processes. Integration is achieved through the use of a uniform message passing sub-system and a common Object Oriented Repository. This allows the various tools to function independently, and permits system deliveries which are restricted to particular sets of tools.



A dataflow diagram editor created using the HyperEdit customisable editor generator.

Research Focus

The HyperCASE suite of tools was chosen to significantly enhance development productivity, by addressing those parts of systems development which are poorly defined and lack high quality tool support. The areas chosen were;

- **Requirements Capture (SODA):**

Obtaining a precise statement of the customer's requirements is a major problem. A tool for recording system requirements, and for dealing with re-use of design and the extraction of design information is being built. This allows input in restricted natural language, and allows design data to be extracted.

- **Automatic Project Tracking (PROTRACT):**

A tool which maximises the amount of automatic project tracking which can be achieved, reducing managers' reliance on error prone human recording, has been built. This tool monitors developers tool usage, and the evolution of project documents. Managers develop and record a project plan, and the systems reports project progress against it.

- **Design Recording (DETRACT):**

Software and systems maintenance are major issues, are difficult and consume large parts of IT budgets. This tool allows the "design decisions" to be recorded, and to be used to allow designers and maintainers to understand the evolving code. Design decisions, or "reasoning", are recorded using a restricted natural language interface. This can be queried, also using a restricted natural language to uncover relationships between decisions, and items such as code, data items and reports.

- **Diagramming Systems (HyperEdit):**

Producing diagram editors is a costly and difficult process. A tool has been developed which is capable of generating new diagram editors which might be used required for

different development methodologies. This tool allows editors to be generated graphically, and to alter the standard representations of diagram symbols.

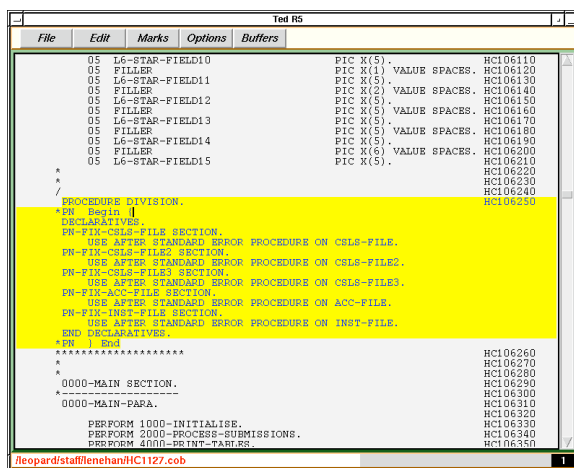
• **The Nature of Software Engineering:**

While not related to our tool development, this research addresses problems that exist in the development of Software Engineering. This research aims at identifying the aspects of engineering which can genuinely be used to influence and improve software development. It has produced many important contributions to our work as a whole.

Hypertext

The complete HyperCASE system is integrated under a Hypertext regime. Tools must therefore be “Hypertext Aware” that is, must be able to both define Hypertext linkages between various documents, and to follow links. Following a link may lead to a new data presentation within the current document by the current tool, or it may lead to the presentation of some part of a different document by either the current or another tool. If necessary new processes are invoked to execute the required tools.

The Hypertext facilities of each tool differ slightly, so only the most immediately interesting are presented here.



TED is a “Hypertext Aware” Text Editor. Source and destination anchors can be defined and linked to any other HyperCASE documents.

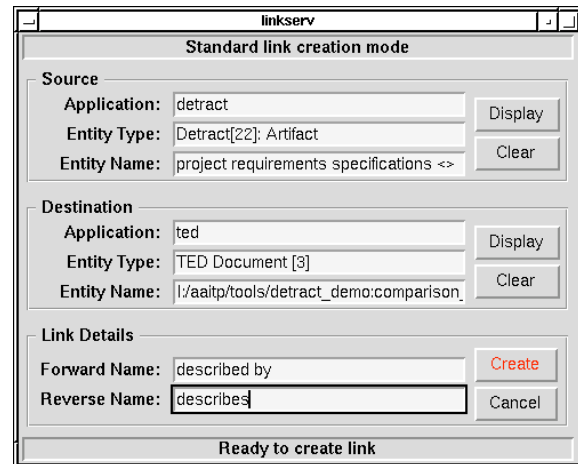
DETRACT, **PROTRACT** and **SODA** documents can serve as link destinations with the relevant tools being invoked to display those documents

as required. In addition various fields within these documents serve as Hypertext link sources and can be linked directly to other project documents.

Within **HyperEDIT** any graphical item (i.e. diagram component, connector, annotation etc.) may be a Hypertext anchor, either a source or a destination. These anchors can be linked to any other project documents.

Within the **Text Editor (TED)** a text chunk of any granularity can serve as a hypertext source or destination anchor.

Link creation is achieved via a highly user-friendly *point and click* mechanism for defining Hypertext source and destination anchors and for forging the links. This mechanism can even be utilised when defining links between documents or data displayed with different tools.



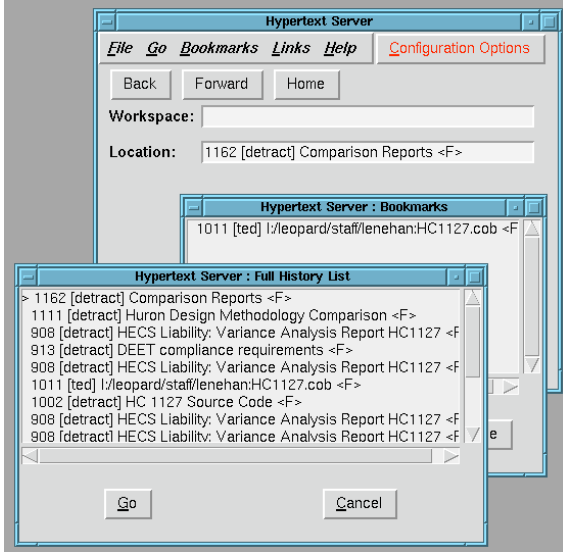
The Link Server window displays hypertext source and destination anchors as they are defined via the user-friendly point and click mechanism.

Implementing HyperCASE as a collection of coupled Hypertext aware tools instead of a single Hypertext display tool provides a straight forward approach for adding hypertext links to external data and even for integrating new or existing third party development tools into the HyperCASE system.

The combination of Hypertext and message based communication approaches can be utilised to provide document animation.

Navigation

Navigation is a major issue for what will be a very large collection of documents and “chunks” (the sub-sections of documents that will be displayed by a HyperCASE presentation tool).



The Hypertext Navigator displays a user's history list and bookmarks set across an entire HyperCASE document collection.

A navigation system is being developed. This makes use of the “natural” structure and evolution of document collections, and allows the user to have different “views”. Special emphasis has been placed on ease of use, and upon combining graphical representations with appropriate query interfaces.

HyperEDIT will be used as a display service for navigation and history maps.

Architecture

HyperCASE is implemented using a "best-practice" architecture. Where ever possible, tools provide common services to other tools, and HyperCASE tools are used in place of non-standard, platform independent components. The loosely coupled client-server message passing architecture maximises the extensibility of the systems, and its capacity to interwork with non-HyperCASE components.

Additional Research

AAITP supports research in a number of areas of Software Engineering in addition to those listed above.

- **Re-engineering:**

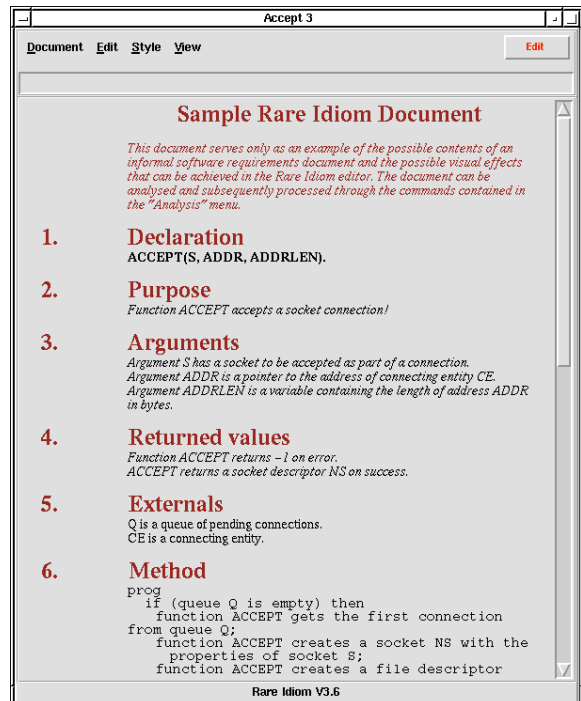
We are investigating the extent to which the HyperCASE tool-set may be used to assist with re-engineering. This has become a major aspect of the work on design recording, where the tools can be used to record a maintainer’s view of the software concerned. Work is also on-going on strategies for actually making changes to existing systems, while maintaining their existing functionality.

- **Software Development:**

The group has a wide interest in software methodologies in general, and is investigating mechanisms for improving these. We are, as part of this activity, interested in Software Process Recording, which will allow actual software practices to be better understood.

HyperCASE Tools and Components

The complete range of the Program can be seen from the following descriptions and list of reports.



The SODA Editor is used to construct and then analyse text documents to identify semantic relationships between components and to highlight component reuse.

- **HyperEDIT** is a customizable graphics editor generator in which each object placed on a screen can be a Hypertext button. It is the centre piece of the AAITP HyperCASE and was proposed by Mr. Cybulski.
- **SODA** is the Software Designers Aid, a system for applying restricted natural language processing concepts to the recognition of common expressions and concepts in documents. The primary objective for this system is Software Requirements capture, however, it may have other uses.
- The **HyperCASE Repository** provides a basis for storing data needed by each tool. The design of this component also affects inter-operability between tools, and, general communications. The underlying database will be a commercial product such as Ingres, Oracle or DBQ. It may be that the Huron™ database interface will be used.
- **PROTRACT** is a project tracking system intended to be coupled to the development environment. It can monitor project progress by tracking tool usage and the evolution of the collection of artefacts making up a software project. It has potential for other applications.

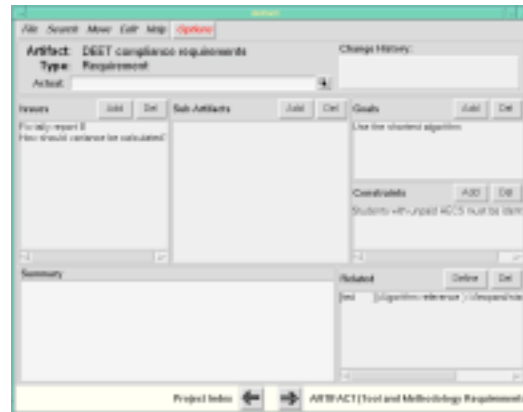


PROTRACT automatically tracks software development activities against a specified plan.

- **Configuration Management.** A technique of managing the configuration of a HyperCASE development project is required. Significant work has already been

done on this, and it can be extended. The objective is to provide an event-based method, in which the configurations are developed from implied references in documents, and, from explicit statements of a 'configuration'. Graphical support for the configurations is needed.

- **DETRACT** is a Design Reasoning Tracking System which can be used to record the reasons for decisions during system development. This will assist maintainers and allow improved communications between team members in a large, multi-team project.



A DETRACT windows for displaying extended artifact documentation including details of related issues, sub-artifacts, goals, and constraints.

- **Graphical Component Based Design.** Development of an improved form of software diagramming system capable of representing a complete design, reflecting the properties of other engineering systems. A major part of the HyperCASE program, this research effort has already yielded significant results in terms of a proposal for a visual language and diagramming system for Huron.
- The **Text Editor (TED)** is a "Hypertext Aware" editor for creating and displaying text documents. Text chunks of any granularity can be defined as Hypertext source or destination anchors. TED is often used to display additional textual annotations for documents or data associated with other HyperCASE tools. These annotations are accessed via Hypertext traversal.
- The **Hypertext Navigator** aids developers as they navigate their way through collections of HyperCASE documents and data. The Navigator provides a traversal history list, the

ability to define bookmarks within the Hypertext, and a tracking map displaying all available Hypertext links from the current document.

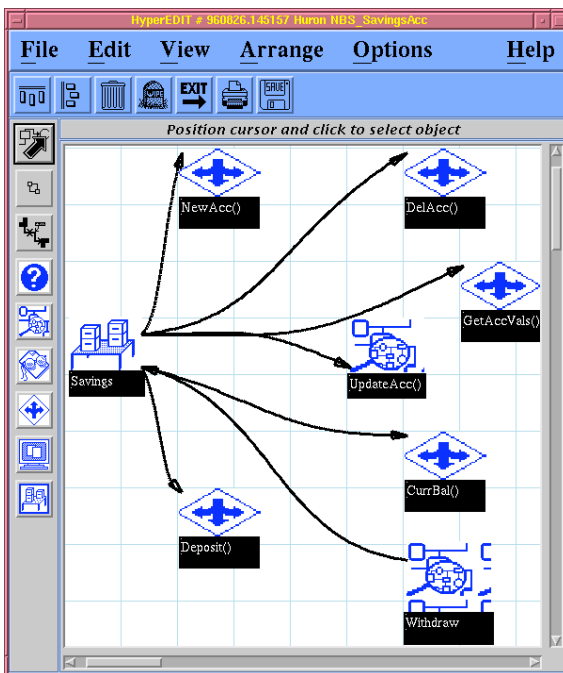
- The **Message Server** is used for all communications between HyperCASE components and tools. This includes repository access, hypertext traversal, and event interception by tools such as PROTRACT, DETRACT, and the Hypertext Navigator. To utilise the message server tools subscribe to particular message classes.

Huron™ Investigations

The AAITP has carried out several investigations into the use and extension of Amdahl's Huron™ development environment.

- **A Diagramming System for Amdahl's Huron™ Development Tool - an Outcome of the AAITP HyperCASE Project** (Baragry and Reed)

An experimental diagramming system for use with the Huron software development system was developed.



A customised HyperEdit editor highlighting the structure of a Huron application.

Huron has a well defined set of building blocks which promotes a high level of component reuse. The diagramming system

allows a developer to use tailored icons to represent Huron elements such as rules, screens, etc. to construct a diagram representing a Huron application. The diagramming system is in fact a combination of diagram notations, which are implemented using the HyperEDIT customizable graphical editor generator.

Collaborative work was performed in 1994 with the Antares Research Institute in Mississauga, Ontario Canada to investigate the integration of HyperCASE diagram systems with the Huron Development Environment.

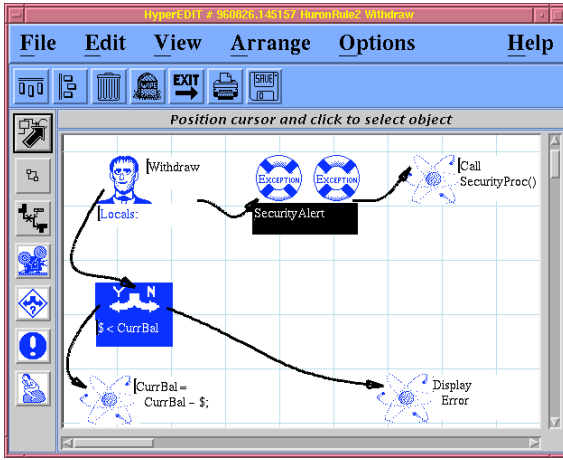
- **Comparisons Between Huron™ and Standard Development Systems** (Cybulski and Reed; 1992)

A number of investigations have been conducted into Huron, and of HyperCASE's capacity for support for this important tool. The approach taken was to design a model application, the "Naive Banking System"¹, (NBS), and to select a series of standard development methodologies. The experimental approach consisted of each experimenter completing a system design in one of the methodologies. The design was then implemented in Huron and in a development system favouring the methodology. The final results are proprietary to Amdahl, and will only be released with their permission. The reports produced are listed below.

- 1) **Investigation of the Rule Modeling Capabilities in Huron™** (Cybulski; 1992; AAITP TR011)
- 2) **The Use of Huron™ as an Object Oriented Development Tool** (Cleary; 1992; AAITP TR015) (Cleary; 1994; AAITP TR034)
- 3) **Comparison of Huron™ with 3GL Design and Implementation using COBOL** (Torabi; 1992; AAITP TR014)
- 4) **Comparison of Huron™ with 4GL Design and Ingres Relational DBMS Implementation** (Proestakis; 1992; AAITP TR013)

¹ The NBS was designed by M. Cybulski.

5) **Comparison of Huron™ with a Hybrid Design and Hypercard/Nexpert Object Implementation**
(Vossos; 1992)



A customised HyperEdit editor providing a graphical display of an individual Huron rule.

The AAITP Team

- **Karl Reed** is an Associate Professor in the Department of Computer Science and Computer Engineering at La Trobe University. Karl is the Director of the AAITP, and its initiator. He is a recognised authority on software engineering and technology policy, and is recognised as national spokesperson on industry policy. Karl's major research interests include software engineering, technology policy and industry structure, computer architecture, and software engineering education. Karl Reed has a total of 32 years experience, of which 12 were in industry. He is Director of the Australian Computer Society's Computer Systems and Software Engineering Board, President of the Board of the International Workshop on CASE, and a member of the Executive of the IEEE-CS Technical Council on Software Engineering. In addition to being a Fellow of ACS and of the Insitution of Engineers Australia, he is also the author of the column "An Essential Reed", which appears in Computer Week. Karl is joint originator of the HyperCASE concept (with Jacob Cybulski) and is the proposer of most of the individual components. He is responsible for the overall conceptual integrity and design of the program and determines HyperCASE's overall directions.

He is directly involved in individual projects through supervision of researchers.

- **David Cleary** is Deputy Director of the AAITP. He is a researcher and is completing his PhD. As its Deputy Director David plays a major role in the day to day management of the AAITP. David's research deals with extending activity progress and status descriptions and automating the tracking of software development projects. This research has led to the development of the PROTRACT tool. PROTRACT automatically tracks software development against software activity descriptions. Activity progress is determined based upon instances and patterns of tool usage and automated analysis of software artifacts and resource consumption data. David was involved with the AAITP's analysis of Amdahl's Huron development system. He carried out the study comparing software development using Huron against development using an object oriented approach and the C++ programming language.
- **Jacob Cybulski** is a senior technical advisor to AAITP . He is currently a Lecturer in the Department of Information Systems at Melbourne University and is completeing a PhD with the AAITP. Jacob is the co-originator of the HyperCASE concept. Jacob was the Deputy Director of the AAITP until the end of 1994. Jacob developed the SODA tool for analysing software documentation expressed in constrained natural language. SODA's analysis identifies and highlights software component references and relationships between components within documents. In addition it is able to identify instances of potential component reuse both within and between documents. This research has since led to the development of a more encompassing reuse framework for software development. Jacob was involved with AAITP's analysis of Amdahl's Huron development system. He carried out most of the initial work in designing the overall comparison framework and performed the individual study comparing software development using Huron against development using a rule based approach and the Prolog logic programming language.

- **Dr John Ophel** is a Lecturer within the School of Computer Science and Computer Engineering at La Trobe University. His research centres on computer programming languages. John serves as a senior technical advisor to the AAITP providing input on a wide variety of issues. These include design of the HyperCASE message based communications system and design of the HyperCASE repository's data model. In addition he has served as joint masters supervisor with Assoc Prof Karl Reed of Mrs Li Li who is working on composite diagramming systems. This work addresses using the HyperEdit tool to construct and display diagrams containing components from diverse diagrammatic methods.
- **Jason Baragry** is a researcher with AAITP and is finishing a PhD. Originally Jason's research addressed the use of diagramming systems and diagram execution in software design. This has since evolved, however, to address issues of component based design in software development and the comparison of software engineering with traditional engineering disciplines. Jason applied these ideas in the development of graphical languages for use in the construction and display of Amdahl's Huron rule components and also to provide a structured graphical model of a Huron component set. Diagrams constructed using these languages are displayed using the HyperEdit tool. The results of Jason's research provide significant inputs and potential solutions for many of the issues dealt with by the AAITP. Jason was involved with the AAITP's analysis of Amdahl's Huron development system, playing a major role in the production of the final report.
- **Torab Torabi** is a researcher with the AAITP. He is currently an Assistant Lecturer with the School of Computer Science and Computer Engineering at La Trobe University, and is an Iranian national scholar. He is also completing a PhD. Torab's research deals with tracking, recording and inference associated with the various design issues that must be dealt with during software development. This has led to the development of the DETRACT tool. Utilising a conceptual model linking the various design issues together, DETRACT is used to record and then play back development documentation that would not be addressed if using conventional techniques. Torab was involved with the AAITP's analysis of Amdahl's Huron development system. He carried out the study comparing software development using Huron against a conventional structured approach and the COBOL programming language.
- **Michael Cooper** is completing a masters by course work with the AAITP. His research addresses the issue of navigation within hypertext systems, with special regard to the document collections which arise during a large-scale software development project. This led to the development of a prototype hypertext navigation control tool which was then re-implemented as an integrated component of the HyperCASE system. This work has implication for hypertext in general.
- **Li Li** is a researcher and is undertaking a masters program with the AAITP. She is also an Assistant Lecturer in the School of Computer Science and Computer Engineering at La Trobe University. Li Li's research addresses the construction and display of diagrams composed of or including components taken from diverse diagrammatic methods. In particular she has dealt with the issues of nested components and connections between components of different types. A major goal of this research is to extend the HyperEdit tool to allow it to display these composite diagrams.
- **Jamie Lenehan** is a principal research programmer with the AAITP. Jamie has worked on detailed design and implementation of a number of HyperCASE components and tools. These include; the HyperCASE message server communication system, the HyperCASE repository, the DETRACT tool's user interface, and a hypertext aware text editor. He has also worked on converting HyperEdit into a product quality tool in terms of performance and reliability, and on the hypertext integration of the various HyperCASE clients though making clients hypertext aware and implementing a *point and click* based

hypertext link creation and modification facility.

- **Fred Brkich** is a research programmer with the AAITP. (Fred is also undertaking a PhD at RMIT.) Fred has worked on the implementation of the natural language processing facility within the DETRACT tool and also upon the implementation of a general purpose query mechanism for the HyperCASE repository.
- **Arthur Proestakis** was a research programmer with the AAITP until mid 1993. During that time Arthur worked in conjunction with Jacob Cybulski in the design of the HyperEdit tool and then implemented the first HyperEdit prototype. This initial work concentrated upon the HyperEdit tool's overall behaviour, its data model, and its graphical user interface. Arthur was involved with the AAITP's analysis of Amdahl's Huron development system. He carried out the study comparing software development using Huron against development using the INGRES relational DBMS and 4GL. Arthur is currently with the Computer Power Group.
- **Mrs Beverley Teague** is the AAITP executive officer. She is responsible for the general management of the Program's extensive set of files, interfacing with La Trobe University's administration and for organising meetings and demonstrations. She handles travel arrangements, organises staff matters and generally provides the Program's interface with the University and the wider community.

AAITP has been host to a large number of honours students, vacations scholars, graduate students and programmers, all of whom have made contributions to its work. These include;

Kevin Alldritt
Andrew Beitz
Teodor Benejariu
Atish Charan
Mel Hatzis
Steve Iatropolous
Daniel Jitnah
Li Li Lim
Albert Ling
Xin Lui
Austin McLaughlin

Megan O'Connor
Micheal O'Keefe
Jopley Oyieng
Jane Philcox
Fadi Salloum
Brett Schubert
Chris Miles
Sam Soubra
Robert van Doorn
Michael Yong

Publications

J. Cybulski, K. Reed, "A Hypertext Based Software Engineering Environment", *IEEE Software*, March 1992.

D. Cleary, K. Reed, "PROTRACT: A Computerised Tool for Tracking Software Development using Event Based Activity Specifications", *Proceedings Australian Software Engineering Conference'93*.

J. Baragry, D. Cleary, T. Torabi, "Unix Inter-Component Communication within a Hypertext Based CASE Tool", *Proceedings Australian Unix Users Group (Victoria) Summer'94*.

J. Baragry, "An Initial Comparison of Software Engineering Design of Automotive Cruise Control Systems", *Proceedings Australian Software Engineering Conference'96*.

Technical Reports

TR001 **K. Reed**, The Original Technical Program for the Amdahl Australian Intelligent Tools Program, 1988.

TR002 **K. Reed**, An Outline of a Knowledge Acquisition Based Approach to Software Project Planning, April 1993.

TR003 **D. Cleary**, A Frame Based Conceptual Scheme for Tracking Document Development in an Extended Hypertext Environment, 1993, Rev.1.1.

TR004 **D. Cleary**, An Automated PROject TRACKing Tool An Initial Software Requirements Specification for Automatic Product "Completedness" Determination, 1991.

TR005 **M. Yong, K. Reed**, Identifying Reusable Components in Software Requirements Specifications to Develop a Natural Language-like SRS Language with a CRNLP, January 1992.

TR006 **M. Hatzis**, Design Considerations of DFDEdit, 1991.

TR007 **A. Proestakis**, HyperEDIT - Functional Description, 1992.

TR008 **J. Cybulski, K. Reed**, HyperCASE: A Hypertext Based Software Engineering Environment, 1991.

TR009 **J. Cybulski, A. Proestakis**, HyperEDIT: An Object-Oriented Diagram Meta-Editor, 1991.

TR010 **J. Cybulski, K. Reed**, Integrating Hypertext and CASE: The Emerging Technology, 1991.

TR011 **J. Cybulski**, Huron: The Rule-Based System, 1992.

TR012 **J. Baragry**, Huron: The Investigation of Advanced Software Development Capabilities of the Huron Development Environment: Results, 1991.

TR013 **A. Proestakis**, Huron: Relational Analysis, 1992.

TR014 **T. Torabi**, Huron: Comparison of Huron and 3GLs, 1992.

TR015 **D. Cleary**, Huron: The Object-Oriented System, 1992.

TR016 **A. McLaughlin**, Software Requirements Specifications for DETRACT A Software Development Design Tracker, 1991.

TR017 **A. Proestakis**, HyperEDIT: Architectural Overview and Distributed Building Process Description, June 1993.

TR018 **J. Cybulski**, SODA: Software Designer's Aide, August 1992.

TR019 **J. Baragry**, A Diagramming System for Amdahl's Huron™ Development Tool. An Outcome of the AAITP HyperCASE

TR020 **D. Cleary, K. Reed**, PROTRACT: A Computerised Tool for Tracking Software, 1993.

TR021 **A. McLaughlin, K. Reed**, Design Rationale: The Missing Ingredient in Software Development, 1991.

TR022 **D. Cleary**, Simple Subscription/Dispatcher for UNIX Interprocess Communication, 1993.

TR023 **D. Cleary**, Using a Decision Table Preprocessor to Improve the Huron Rule Language, 1993.

TR024 **J. Cybulski**, The Use of Templates and Restricted English in Structuring and Analysis of Informal Requirement Specifications, 1993

TR025 **J. Cybulski**, The Impact of Artificial Intelligence on Data Base Development , 1991

TR026 **D. Cleary**, RACV Software Developer Tool Usage Study Summary, November 1993.

TR027 **J. Baragry, D. Cleary, T. Torabi**, Unix Inter-component Communication within a Hypertext Based CASE Tool, 1994.

TR028 **T. Torabi, K. Reed**, "DETRACT: A DEsign TRACKing Tool:, 1993.

TR029 **J. Baragry, K. Reed**, Extending Visual Programming Graphical Component Based Design, April 1994.

TR030 **D. Cleary**, Project Tracking Through Document Status Analysis, 1994.

TR031 **T. Torabi**, DETRACT Relational Repository Manager (DRRM), 1993.

TR032 **T. Torabi**, A Review of SoftEAM and Comparison with DETRACT, November 1993.

TR033 **J. Baragry**, Software Components vs Engineering Components: Observations, January 1994.

TR034 **D. Cleary**, Huron and Object Orientation 1994.

TR035 **M. Cooper**, Hypertext Navigation, A Bibliography, May 1995.

TR036 **X Liu, A. Proestakis**, Software Requirements Specifications for HyperEDIT - A Diagram Editor for HyperCASE, October 1994

TR037 **D. Cleary**, Tracking Activity Occurrence Against an Activity-Ordering Relationship Network, May 1995.

TR038 **D. Cleary, T. Torabi, K. Reed**, Query Sub-System for a Design Decision Tracking Tool, May 1995.