Knowledge Based Reverse Engineering of Legacy Telecommunications Software

Mitel Corporation
University of Ottawa

Progress Presentation
May 1998

Outline:

Key Progress in the Past Six Months

Ideas Under Development

Other Ongoing Work

Lessons Learned from Industrial
NRC and CSER Collaboration

Key Progress in the Past Six Months

1. Extracting clusters

2. Metrics for coupling and cohesion

3. Parsing in the presence of conditional compilation

4. Productization at Mitel

1. Extracting Clusters

Leads: N. Anquetil, S. Somé

Discovered that abbreviations in file names are a good way to find suitable clusters (in Mitel system)

Comparable with clusters generated using:

- Type use similarity
- Data use similarity
- Routine call / called by similarity

Published papers about this in CASCON & ICSE

Submitted paper to WCRE

2. Metrics for cohesion and coupling

Lead: Nicolas Anquetil

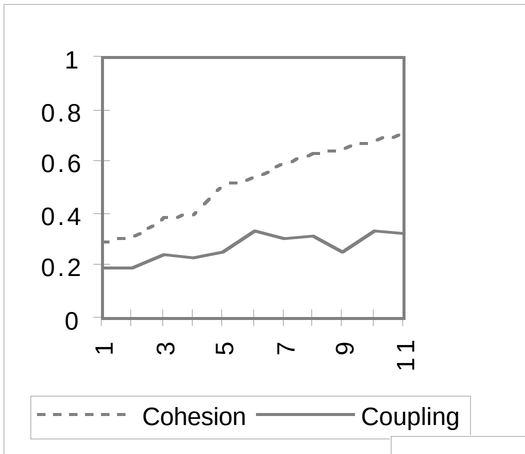
The literature (e.g. Kunz) provides cohesion & coupling metrics

Quality is defined as cohesion minus coupling

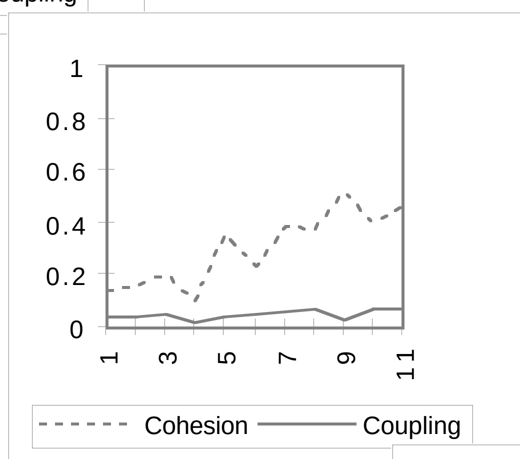
- Unfortunately, almost completely correlated with cohesion
  - because subtracting a small number from a large one
  - see figures on next page

Cohesion and coupling cont.

Literature metrics



Type clustering

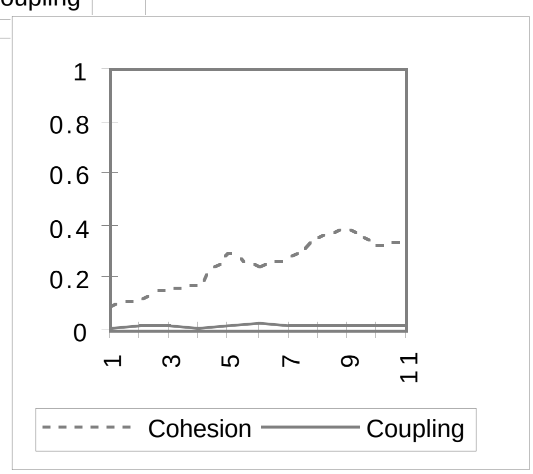


Data clustering

X axis lists 11 methods of generating clusters

(including manual and automatic generation - see next slide)

Routine clustering



The clustering techniques (for reference)

Description of clustering technique	**Clustering technique number** (where 1 means least cohesive and 11 means most, according to figure 1)
Clusters created manually	
Clusters selected by experts (did not include all files)	6
Clusters recorded in the configuration management system	3
Similar to 3, but only considering clusters containing at least one file from the experts' partition (6)	4
Clusters generated automatically .. using similarity based on data references	
Automatic data-reference clustering	7
Similar to 7, but only considering clusters containing at least one file from the experts' partition (6)	11
... using similarity based on routine calls	
Automatic routine-call clustering	8
Similar to 8, but only considering clusters containing at least one file from the experts' partition (6)	10
... using similarity based on abbreviations in file names (Anquetil and Lethbridge 1998)	
Automatic file name clustering	1
Similar to 1, but only considering clusters containing at least one file from the experts' partition (6)	5
Similar to 1, but only considering first abbreviation in each file name	2
Similar to 2, but only considering clusters containing at least one file from the experts' partition (6)	9

Cohesion and coupling cont.

We defined upper/lower bounds on cohesion and coupling of any subsystem in a system

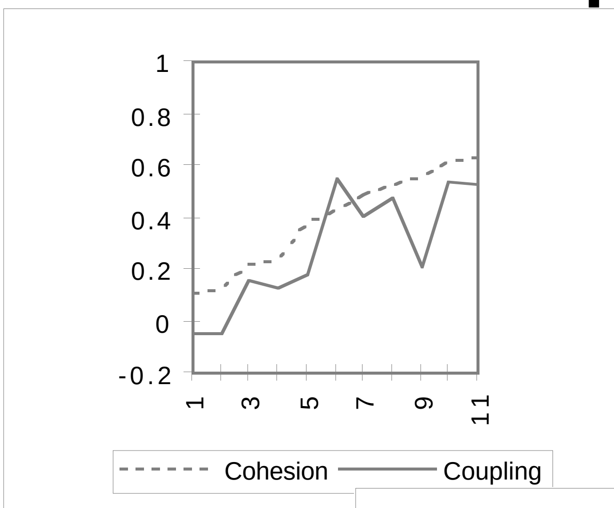
Used this to define normalized metrics

- Allows metric values to range from 0 - 1
- Results in consistent quality metric
  - Cohesion and coupling contribute equally

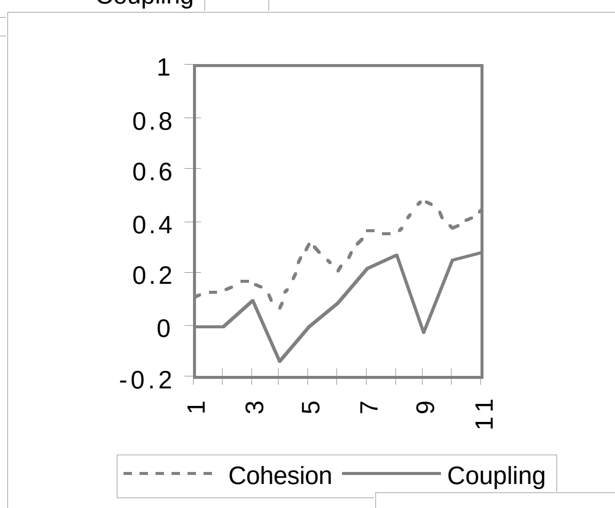
Submitted to metrics symposium

Cohesion and coupling cont.

Normalized metrics

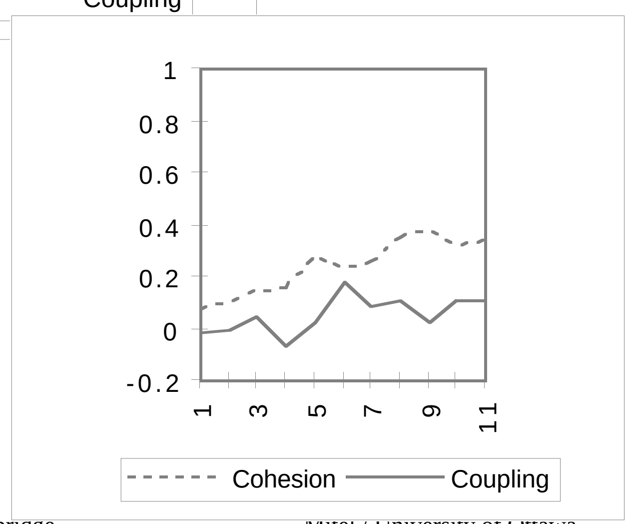


Type clustering

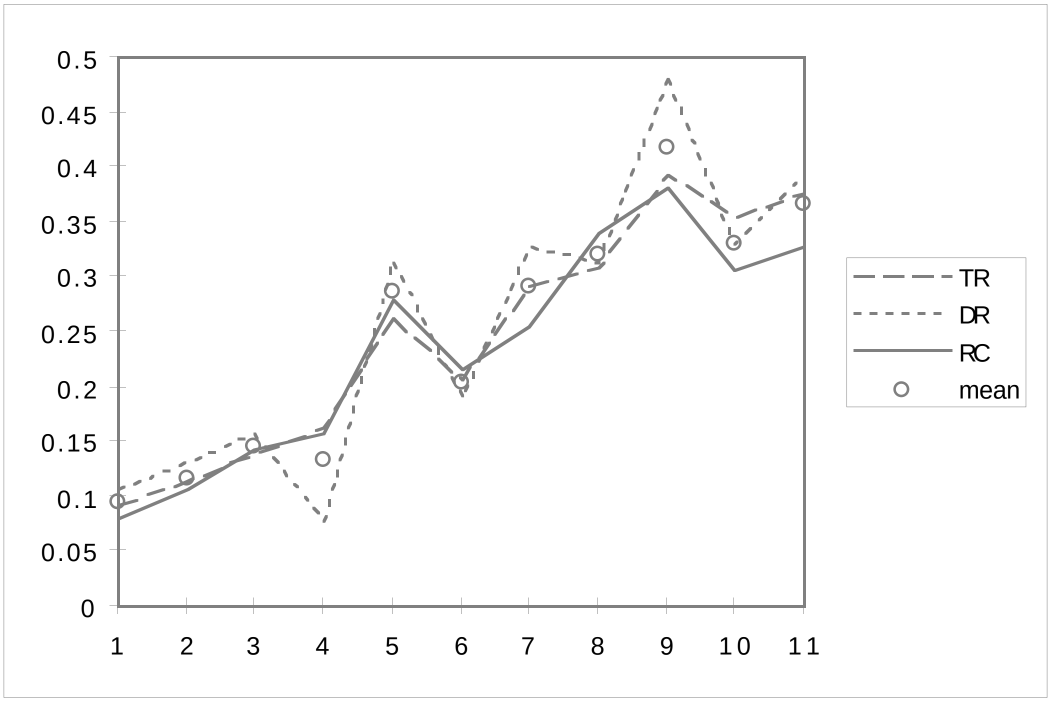


Data clustering

Routine clustering

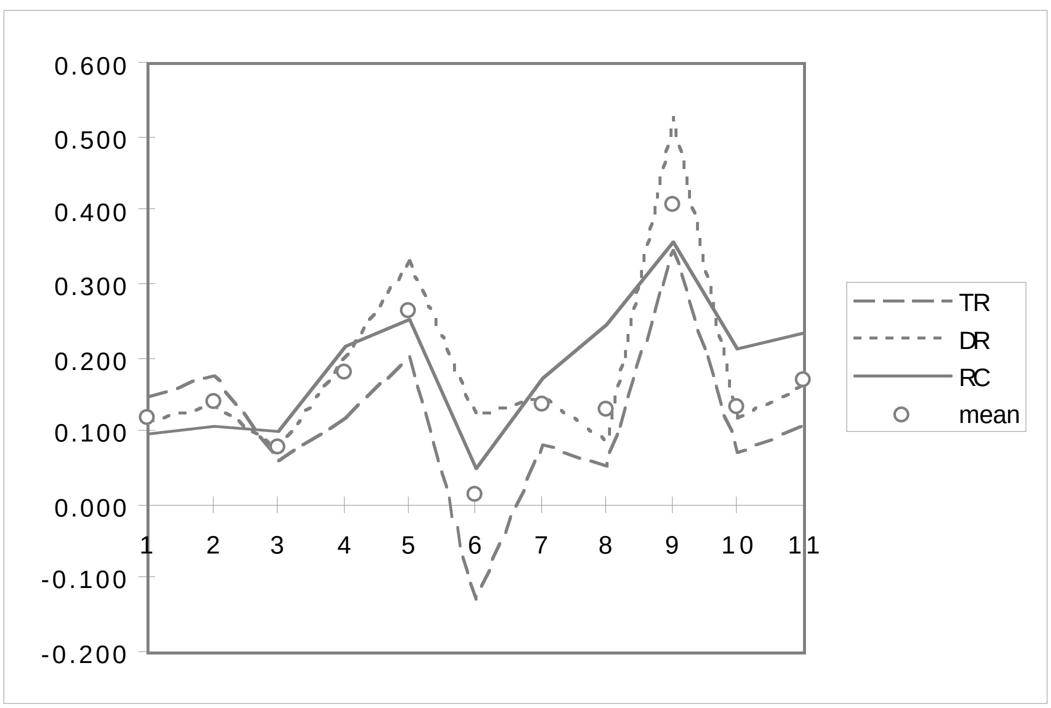


Cohesion and coupling cont.



Quality

Normalized quality



3. Conditional compilation parsing problem

Lead: Stéphane Somé

Paper at IWPC based on research reported at CASCON CSER meeting

4. Productization at Mitel

Jerry Chen has taken over routine work needed to ensure quality within Mitel

Modifications of the tool for research purposes continues in parallel

Ideas Under Ongoing Development

5. Vision for hierarchy browsing of subsystems

6. Machine learning to assist browsing

7. Browsing of states, processes etc.

8. 1998 education relevance survey

5. Vision for hierarchy browsing of clusters

Key idea:

- Extend Just-In-Time Comprehension (JITC) to work with clusters

Current UI uses hierarchy of hierarchies paradigm

- Outer: Exploration states (history)
- Inner: Places visited in exploration
  - Files, routines, types, fields etc.

Cluster browsing cont.

Extension to work with clusters:

- New exploration node: The *cluster*
  - Created by user when doing JITC
  - Or created by clustering algorithm
- Operations on any set of selected objects
  - Differences / similarities / interconnections

    Works on *members* of selected clusters
  - Group into cluster
  - Drop from view
  - Focus on these alone
- When 1 or more clusters is selected:
  - Show members
  - Name the cluster
- When 2 or more clusters selected:
  - Intersection / set difference
  - Differences between members

6. Machine learning to assist browsing

Lead: Jelber Sayyad Shirabad

Each step of exploration recorded

- When you look at X, what do you typically look at subsequently?
- Three classes:
  - Not looked at
  - Selected, but not explored
  - Explored
- Use machine learning to build ‘relevance rules’
- Result: Intelligent assistance
  - When future users browse similar code, system suggests what might be useful to look at

7. Browsing of states, processes etc.

Lead: Stéphane Somé

Currently: Database only contains information extracted from syntax

- Allows SE’s to browse along certain architectural dimensions (ADs)
  - Calls-called by
  - Data/type usages ADT’s/classes
  - Clusters / subsystems
- System independent

States etc. cont.

Other ADs require system ‘knowledge’

- States:
  - How are states represented in this system?
  - Procedures? Variables? Enumerated types?
- Platforms / devices / etc.

Idea:

- Create ‘rules’ / ‘parameters’ for each system that describe major architectural dimensions
- JITC can then be performed along these dimensions:
- e.g.
  - ‘Show me the states from which this routine can be called’
  - ‘Show me differences between these states’

8. 1998 education relevance survey

169 participants in 1997

Presented at IEEE Conference on Software Engineering Education and Training

- Enthusiastically received
- Detailed paper under review for Annals of SE

Plans for a 1998 survey

- Improved questions
- Collaboration with researchers Georgia Tech, RIT, Software Engineering Institute, Universities in the UK etc.
- Goal: 1000 participants

Other Ongoing Work

9. Improving knowledge about what software engineers do

- Tim Lethbridge / Janice Singer
- Interviewing and synchronized shadowing of software developers
- Analyzing usage of tools (standard tools and tkSee)
- Guiding tool-development efforts

10. Performing survey on field-study methods in software engineering

- Categorizing field studies and their methods in the literature
- Developing field study guidelines
- Hope to publish in IEEE Transactions on Software Engineering

Other Ongoing Work cont.

11. Providing more sophisticated data model and queries in tkSee
  - Steven Yandon
  - Installing new database infrastructure at Mitel
  - Cutover when stable and fast

12. Studying correlation between concepts extracted from source code comments and concepts identified by experts
  - Jelber Sayyad
  - Possible source of information for browsing

Other Ongoing Work cont.

13. Broadening database coverage to other programming languages
- Assembler: Cedric Fourier
  - Widely used in Mitel

Lessons Learned From Industrial Collaboration

People are very willing to work with you as long as:

- They see some tangible benefit (does not have to be big)
- You respect their need to do their work
- Support and enthusiasm of management is present

Managers and engineers have been inspiring

- New ideas, needs, criticisms etc.

Lessons (Industrial) cont.

Can’t expect to adhere rigidly to milestones

- May suspend or drop milestones as new ideas surface
- Ideas / interests of team members may not match perfectly with plans
  - More productive to let people follow interests than force them to follow plan
- Nevertheless milestone planning helps define objectives
- Probably should not be called milestones

Long term commitment from both parties essential

- Gratifying to have continued support despite realignment of plan details

Lessons (Industrial) cont.

You have to expect staff turnover

- University and company staff
- Has caused significant delays

Hard to find grad students / university staff who want to do field studies / travel to location

Need for ethics approval for work with software engineers

- Informed consent
- Managers must not know who has said know, is doing well, is doing badly

Lessons From CSER-Wide Collaboration

Collaboration between teams with compatible technologies is fruitful

Common theme work hard to ‘get to’ in short period since ‘local’ problems take precedence

- May work well on a longer time horizon
- Should periodically review possible common themes

Lessons from NRC Collaboration

Excellent collaboration with Janice Singer

- Has been major factor in project success

Have used demo-centre hardware remotely since we needed large amount of CPU power