



Software Improvement Group



Comparative Study of Code Query Technologies

Tiago L. Alves (SIG & Universidade do Minho)

September 2011

T +31 20 314 0950

info@sig.eu

www.sig.eu



Software Risk Assessment

- In-depth investigation of software quality and associated business risks
- Answers to specific research questions



Software Monitoring

- Continuous measurement, feedback, and development consultancy
- Guard quality from start to finish



Software Product Certification

- Five levels of technical quality (maintainability)
- Evaluation by SIG, certification by TÜV Informationstechnik

Who is using our services?



Software Improvement Group

Financial and Insurance companies

Government

Logistical

IT

Other

Financial and Insurance companies:

- ABN-AMRO
- ING
- Rabobank
- achmea
- Friesland Bank
- LeasePlan
- Interpolis
- Bank Mendes Gans
- globalcollect™
- Allianz
- PGGM
- InterBank
- Zorg en Zekerheid
- GENERALI verzekeringsgroep
- delta lloyd
- Eurobank EFG
- ZwitserLeven
- KAS BANK
- SNS Bank
- Volvo OtoFinans

Government:

- Rijksoverheid
- kadaster
- Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
- Belastingdienst
- Raad voor Rechtsbijstand
- POLITIE
- CWI
- S V B
- RDW

Logistical:

- DHL
- KLM
- TNT
- EUROMAXX TERMINAL
- norfolkline
- ProRail
- Port of Rotterdam

IT:

- Getronics PinkRocade
- CENTRIC
- Capgemini
- Exact® software
- IBM+
- CHSS
- logica

Other:

- ENECO energie
- PRICEWATERHOUSECOOPERS
- essent
- SWISSLEX
- KPMG
- NXP
- Gasunie
- TNO
- IKEA
- Alcatel-Lucent
- Electrabel

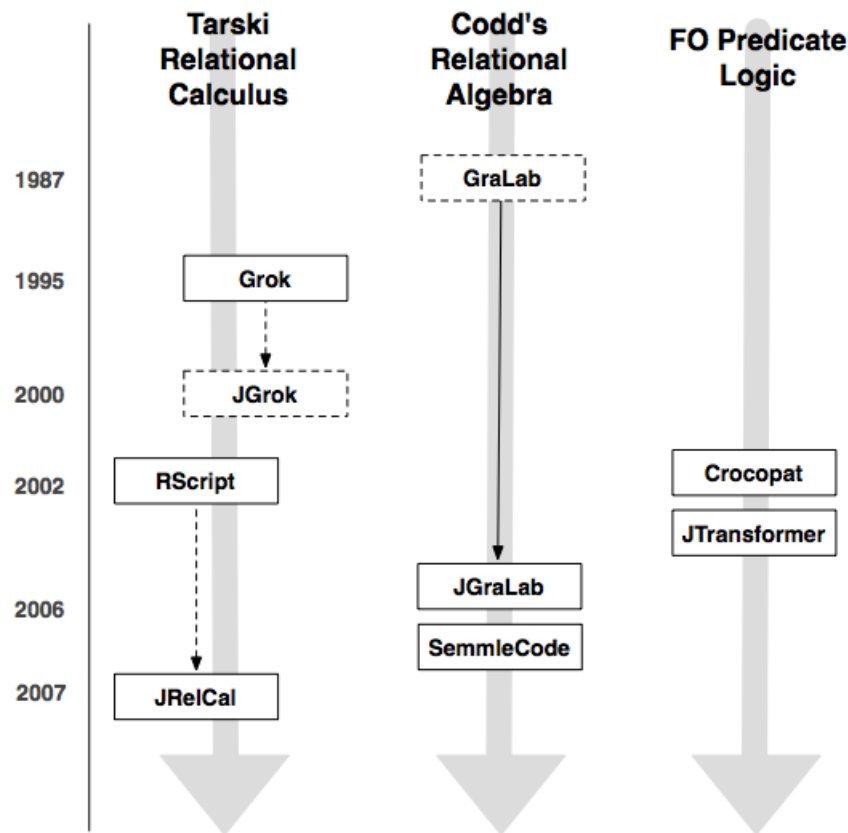


- **Current implementation of SIG tooling**
 - Extract: graph to store facts for several languages
 - Abstract/enrich: implemented using Java visitors
 - Present: through tables and charts
- **Problems**
 - Implementation verbose and imperative
 - Reuse among analyses difficult
 - Error prone
- **Use of code query technologies to improve SIG developer's productivity**
 - Replace current imperative implementation by a more declarative one

Code query languages



Software Improvement Group



Grok, JGrok

- Ric Holt, Canada
- Implemented in Turing

Rscript

- Paul Klint, Netherlands
- Implemented in ASF+SDF

JRelCal

- Tijs van der Storm, Netherlands
- Implemented in Java

GraLab, JGraLab

- Jürgen Ebert, Germany
- Implemented in Java

SemmleCode

- Oege de Moor, UK
- Implemented in Java

CrocoPat

- Dirk Beyer, Germany
- Implemented in C

JTransformer

- Günter Kniesel, Germany
- Implemented in Java

Comparison (not competition)



Software Improvement Group

6 | 17

Code query example

- Experience the language and tool

Language criteria

- Overview of the language features

Tool criteria

- Overview of the tool features

Style/Paradigm

- Compare implementation conciseness

Types

- Support for Integers, Chars, Strings, ...

Parameterization

- Behavior depends on a parameter value

Polymorphism

- Abstract over the entities types

Modularity

- Reuse of queries to construct other queries

Libraries

- Support for libraries of queries

Tool criteria



Software Improvement Group

8 | 17

Output formats

- Text, preformatted text, tables, charts, others?

Interactive interface

- Command line interface (CLI), Graphical user interface (GUI), Eclipse plug-in

API support

- Invocations of the functionality from a host program

Interchange format

- To store facts from the extraction and results of abstraction

Extraction support

- None, Java, C/C++, XML, others?

Licensing

- Free, Open-source, Proprietary



Interactive use

- The tool is used directly by the software analyst (exploratory)
- The user specifies and executes the queries, and extracts results

Tool integration

- The tool is used by a programmer as a component to build other tools
- Reimplementation of existing functionality

Criteria vs. Scenarios



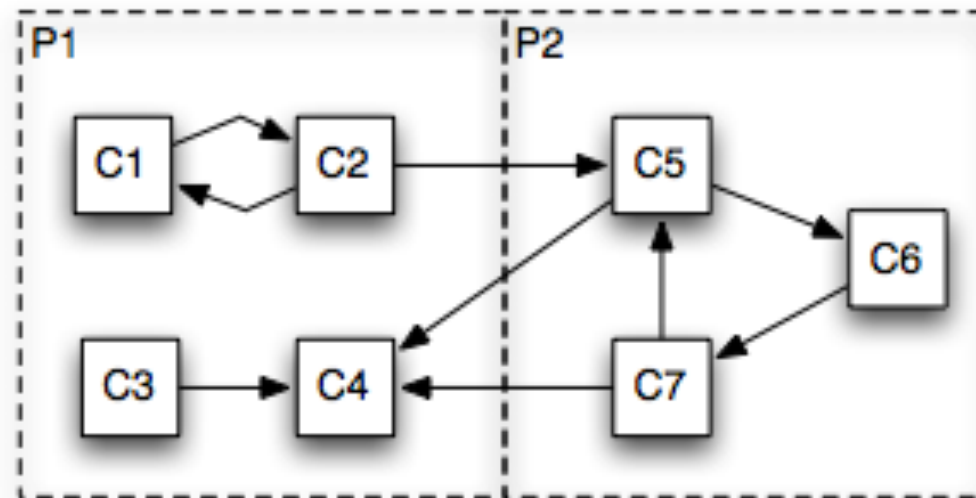
Software Improvement Group

Criteria vs. Scenarios		Interactive use	Tool integration
Language	Style/Paradigm	Important	Important
	Types	Important	Relevant
	Parametrization	Important	Relevant
	Polymorphism	Important	Relevant
	Modularity	Important	Relevant
	Libraries	Important	Relevant
Tool	Output formats	Important	Important
	Interactive use	Important	Not important
	API support	Not important	Important
	Interchange format	Important	Important
	Extraction support	Important	Relevant
	Licensing	Relevant	Important

Code query example - Package instability



Software Improvement Group



11 | 17

Afferent Coupling

C_a = # of classes outside the package that depend upon classes within the package

Efferent Coupling

C_e = # of classes inside the package that depend upon classes outside the package

Package Instability $I = C_e / (C_a + C_e)$

Abstraction

Reachability

Metrics computation

Example comparison



Software Improvement Group

12 | 17

See excerpts in the paper

Examples could be implemented in most tools

- Grok: not possible to fully implement Package Instability
- Rscript: changed computation (no support for reals – multiply result by 100)
- Crocopat: print out the results and read them afterwards
- JGraLab: store results in a variable and use them for the following computations
- SemmleCode: use of “class-less” predicates

Language comparison



Software Improvement Group

Criteria vs. tools		Grok	Rscript	JRelCal	SemmlCode	CrocoPat	JGraLab	JTransformer
Style/paradigm		Relational	Relational & comprehensions	API Relational	OO & SQL-like	FO-logic & imperative	SQL-like & Path expr.	FO-logic
Types	String	x	x	x	x	x	x	x
	Int	x	x	x	x	x	x	x
	Real	x	-	x	x	x	x	x
	Bool	-	x	x	x	x	x	x
	Other	-	Composite & location	Java	Class	-	Edge & Node	Logic terms
Parametrization		-	x	x	-	x	x	x
Polymorphism		-	x	x	x	-	x	x
Modules		x	x	x	x	x	-	x
Libraries		-	-	x	x	-	-	x

Tool comparison



Software Improvement Group

Criteria vs. tools	Grok	Rscript	JRelCal	SemmlCode	CrocoPat	JGraLab	JTransformer
Output formats	Text	Rstore	Sets & Relations	Text, Charts, maps, graphs	Text, RSF	Text, HTML	Text
Interactive interface	CLI	CLI, GUI	-	CLI, Eclipse	CLI	CLI	Eclipse
API support	-	-	x	x	x	x	x
Interchange format	RSF, TA	Rstore	RSF	-	RSF	TGraph	Prolog
Extraction support	C++	-	-	Java, XML	-	Java, C	Java
Licensing	-	BSD	LGPL	Proprietary	LGPL	GPL 2 Proprietary	EPL

Summary



Software Improvement Group

15 | 17

Language criteria

- No significant differences found
- It is not possible to implement Package Instability in Grok

Tool criteria

- Significant differences: interchange format, extraction, licensing
- Poor support for extraction

Interactive use

- Only JRelCal is less suitable.

Tool integration

- JRelCal, SemmleCode, CrocoPat, JGraLab, JTransfomer
- Grok, Rscript only through interchange format

Conclusion



Software Improvement Group

16 | 17

Compared seven code query technologies

- Package instability example
- Six language criteria
- Six tool criteria

Comparison not evaluation

Presented findings

- Allow an informed decision about which tool to choose

Future work & challenges



Software Improvement Group

17 | 17

Future work

- Add more tools / formalisms
- Performance comparison

Challenges

- Adoption of each tool stronger points
- Better support for libraries, interchange format and extractors
- Availability of API
- Interfacing through IDE

Research directions

- Analyze several versions of software
- Architecture checking



Software Improvement Group

18 | 17

Thank you
t.alves@sig.eu