CERN-ITC Code Analysis

Alessandra Potrich

ITC-Irst, Centro per la Ricerca Scientifica e Tecnologica

potrich@itc.it

Outline

- Coding conventions: progress status
- Filtering: a new feature added
- Reverse engineering tool: the first prototype

Coding conventions

- more rules have been implemented (RC7, RC10, RS2 and RS3)
- implementation of some others (RC6, RC12, RN3) has been improved

..... summarizing

Conventions	Abs.	Perc.
Total	46	
Implementable	34	74%
Implemented	30	88%

State of development

Conventions	Total	Impl.	To be Impl.	Non Impl.	Excl.
Naming Rules	21	17	0	3	1
Coding Rules	14	10	1	1	2
Style Rules	5	3	2	0	0
Naming Guidelines	2	0	0	2	0
Coding Guidelines	4	0	1	3	0
Style Guidelines	0	0	0	0	0

Filtering

- why?
  - primarily, to exclude code generated automatically by ROOT;
  - but also, to focus the analyses;
- how?
  - by removing the specified entities before rules checking;
  - configured through a suitable file ("config_FILTERING");

... an example

```
MODULE_NAME
GLOBAL_VARIABLE_NAME %R__
CLASS_NAME %R__
METHOD_NAME DeclFileName DeclFileLine ImplFileName
FIELD_NAME fgIsA ^f $Generator
LOCAL_VARIABLE_NAME
FRIEND_FUNCTIONS_NAME operator>>
```

		old	ROOT filt. OFF	ROOT filt. ON	+ structures filt.
RN3	No _,#,$,&,,,%	1171	859	481	469
RN4	Header file = Classname.h	15	15	15	14
RN5	Implementation file names = Classname.cxx	15	15	15	14
RN6	Class names start with the prefix Ali	38	38	38	11
RN9	Member function names start with a capital.	3	3	3	3
RN11	Data member names start with a prefix "f"	396	396	396	81
RN13	Local variables start with a lower case letter.	599	599	581	581
RN15	Global variables start with a prefix "gAli"	260	260	87	87
RN17	Constants start with a prefix "k"	986	370	370	370
RN19	Static data members start with a prefix "fg".	13	13	13	13
RC3	Multiple-inclusion protection	107	107	107	107
RC4	Only one class in header file	23	23	23	22
RC5	Only one class in implementation file	32	32	32	32
RC6	Order of public, protected and private in class	173	173	67	67
RC7	Global variables	–	128	27	27
RC8	Friend classes	0	0	0	0
RC9	Virtual destructor	31	31	10	10
RC10	Copy constructor and an assignment operator	–	178	69	69
RC12	Dummy argument in member function declarations	199	123	123	123
RC14	Data members "private" or "protected"	545	545	545	228
RS2	Field comment	–	785	614	309
RS3	Method comment	–	2101	1213	1213
RS5	"inline"	17	17	17	17

ROOT filtering ON

Rules		ALIROOT	CASTOR	EVGEN	FMD	ITS	MUON	PHOS
RN3	No _,#,$,&,,,%	1		8		4	98	119
RN4	Header file = Classname.h			2		1	7	1
RN5	Implementation file names = Classname.cxx			2		1	7	1
RN6	Class names start with the prefix Ali					1	3	2
RN9	Member function names start with a capital.							
RN11	Data member names start with a prefix "f"			26		12	22	37
RN13	Local variables start with a lower case letter.		4	34	6	10	162	119
RN15	Global variables start with a prefix "gAli"	5					19	2
RN17	Constants start with a prefix "k"			52	4	21	93	33
RN19	Static data members start with a prefix "fg".					12		
RC3	Multiple-inclusion protection		1	11	3	14	15	4
RC4	Only one class in header file		1	2	1	1	4	2
RC5	Only one class in implementation file						10	
RC6	Order of public, protected and private in class		1	12		13	5	3
RC7	Global variables	1					6	1
RC8	Friend classes							
RC9	Virtual destructor						3	
RC10	Copy constructor and an assignment operator			9		10	12	1
RC12	Dummy argument in member function declarations		1	17	2	7	34	1
RC14	Data members "private" or "protected"		1	1	1	22	73	39
RS2	Field comment			94		17	72	38
RS3	Method comment		8	124	9	84	225	106
RS5	"inline"							

ROOT filtering ON

Rules		PMD	RALICE	RICH	START	STEER	STRUCT	TGeant3
RN3	No _,#,$,&,,,%	20	4	38	2	14	40	85
RN4	Header file = Classname.h			3				1
RN5	Implementation file names = Classname.cxx			3				1
RN6	Class names start with the prefix Ali			2				27
RN9	Member function names start with a capital.			2				1
RN11	Data member names start with a prefix "f"			6				278
RN13	Local variables start with a lower case letter.	2	9	45	5	13	21	
RN15	Global variables start with a prefix "gAli"	44		4		7		5
RN17	Constants start with a prefix "k"	8	11	16		49	6	8
RN19	Static data members start with a prefix "fg".							
RC3	Multiple-inclusion protection	4		7	2	10	15	2
RC4	Only one class in header file	1		4	1	1		1
RC5	Only one class in implementation file			22				
RC6	Order of public, protected and private in class	4		3		12		1
RC7	Global variables	3		4		5		2
RC8	Friend classes							
RC9	Virtual destructor		2	3		2		
RC10	Copy constructor and an assignment operator		7	10		9		1
RC12	Dummy argument in member function declarations	6	1	11	1	10	1	11
RC14	Data members "private" or "protected"	2		43	6	5		303
RS2	Field comment	3		42	3	5		309
RS3	Method comment	17	4	141	7	163	36	47
RS5	"inline"	4				13		

ROOT filtering ON

Rules		TOF	TPC	TRD	ZDC
RN3	No _,#,$,&,,,%	1	41	5	1
RN4	Header file = Classname.h				
RN5	Implementation file names = Classname.cxx				
RN6	Class names start with the prefix Ali		3		
RN9	Member function names start with a capital.				
RN11	Data member names start with a prefix "f"		15		
RN13	Local variables start with a lower case letter.	5	117	15	14
RN15	Global variables start with a prefix "gAli"		1		
RN17	Constants start with a prefix "k"	3	51	11	4
RN19	Static data members start with a prefix "fg".		1		
RC3	Multiple-inclusion protection	5	7	6	1
RC4	Only one class in header file	1	1	1	1
RC5	Only one class in implementation file				
RC6	Order of public, protected and private in class	5	5	3	
RC7	Global variables		4	1	
RC8	Friend classes				
RC9	Virtual destructor				
RC10	Copy constructor and an assignment operator		8	2	
RC12	Dummy argument in member function declarations	5	11	3	1
RC14	Data members "private" or "protected"	6	30	11	2
RS2	Field comment	1	18	12	
RS3	Method comment	12	168	55	7
RS5	"inline"				

ROOT filtering ON

Rules	A	C	E	F	I	M	PH	PM	RA	RI	STA	STE	STR	TG	TO	TP	TR	Z	TOT
RN3	1		8		4	98	119	20	4	38	2	14	40	85	1	41	5	1	481
RN4			2		1	7	1			3				1					15
RN5			2		1	7	1			3				1					15
RN6					1	3	2			2				27		3			38
RN9										2				1					3
RN11			26		12	22	37			6				278		15			396
RN13		4	34	6	10	162	119	2	9	45	5	13	21		5	117	15	14	581
RN15	5					19	2	44		4		7		5		1			87
RN17			52	4	21	93	33	8	11	16		49	6	8	3	51	11	4	370
RN19					12											1			13
RC3		1	11	3	14	15	4	4		7	2	10	15	2	5	7	6	1	107
RC4		1	2	1	1	4	2	1		4	1	1		1	1	1	1	1	23
RC5						10				22									32
RC6		1	12		13	5	3	4		3		12		1	5	5	3		67
RC7	1					6	1	3		4		5		2		4	1		27
RC8																			0
RC9						3			2	3		2							10
RC10			9		10	12	1		7	10		9		1		8	2		69
RC12		1	17	2	7	34	1	6	1	11	1	10	1	11	5	11	3	1	123
RC14		1	1	1	22	73	39	2		43	6	5		303	6	30	11	2	545
RS2			94		17	72	38	3		42	3	5		309	1	18	12		614
RS3		8	124	9	84	225	106	17	4	141	7	163	36	47	12	168	55	7	1213
RS5								4				13							17

The Reverse Engineering Tool

Requirements:

- `java library grappa` (to visualize graphs)
- `dot` (to compute graphs layout)

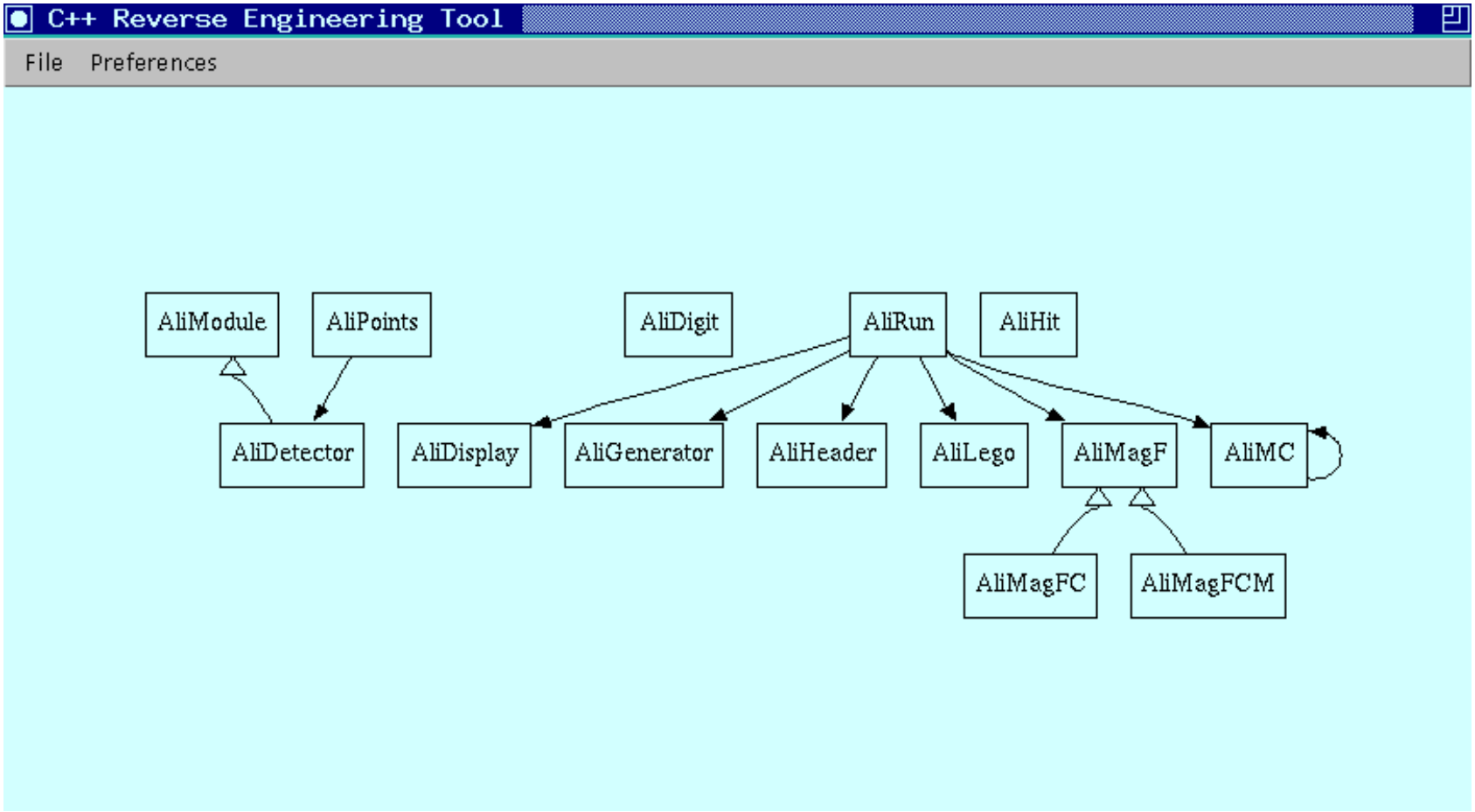
Use:

```
> java reveng.ReverseEngineering *.ii
```

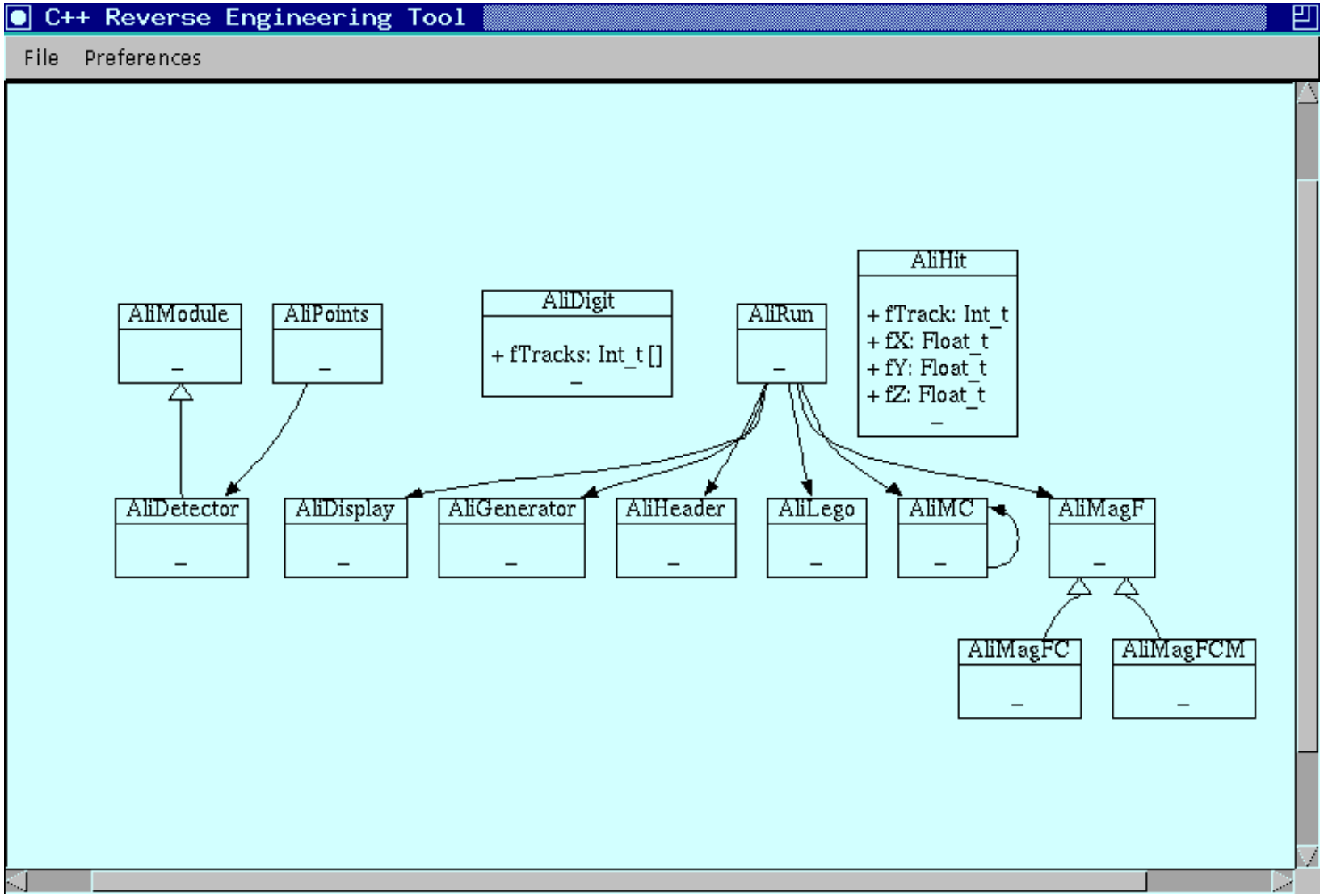
Output:

- in a (default) file named *classDiagram.dot*

Visualization




```
> java reveng.RevEngInterface classDiagram.dot
```


C++ Reverse Engineering Tool
File
Preferences
AliModule
AliPoints
AliDigit
+ fTracks: Int_t []
AliRun
AliHit
+ fTrack: Int_t
+ fX: Float_t
+ fY: Float_t
+ fZ: Float_t
AliDetector
AliDisplay
AliGenerator
AliHeader
AliLego
AliMC
AliMagF
AliMagFC
AliMagFCM