CodeCompass

an Open Software Comprehension Framework

Zoltán Porkoláb[1,2], Dániel Krupp[1], Tibor Brunner[2], Márton Csordás[2]

https://github.com/Ericsson/CodeCompass

Motto: If it was hard to write it should be hard to understand

-- unknown programmer

[1]Ericsson Ltd, [2]Eötvös Loránd University, Budapest, Hungary

Agenda

- Comprehension as a cost factor
- Why development tools are not perfect for comprehension?
- Requirements
- Architecture
- A few workflows
- Restrictions
- Experiences
- Further planes

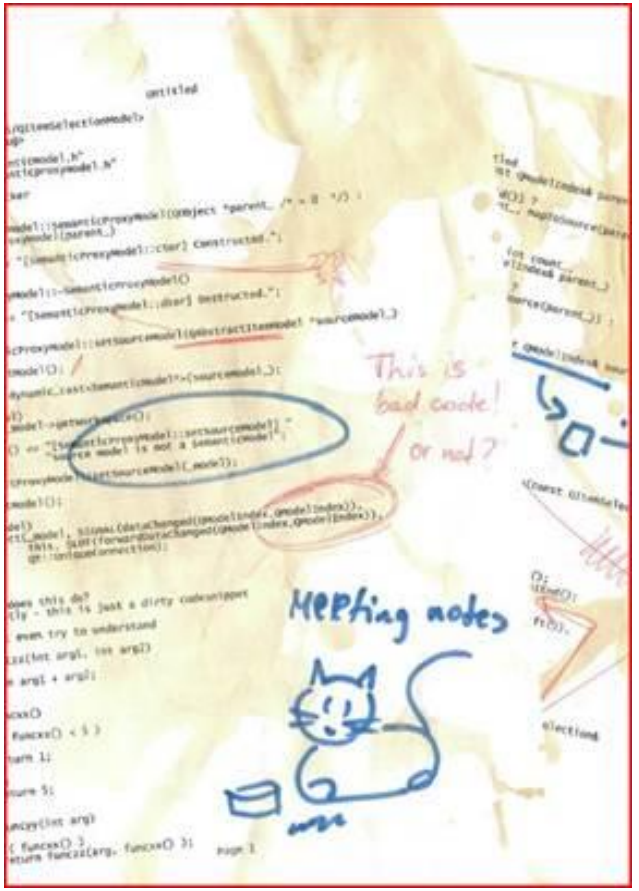
Comprehension is a major cost factor

Research	Effort for comprehension
IBM (Corbi, 1989)	Over 50% of time
Bell Labs (Davison, 1992)	New project members: 60-80% of time, drops to 20% as one gains experience
National Research Council in Canada (Singer, 2006)	Over 25% of time either searching for or looking at code
Microsoft (Hallam, 2006)	Equal amount of time as design, test
Microsoft (La Toza, 2007)	Over 70% of time
Microsoft (Cherubini, 2007)	95%~ significant part of job 65%< at least once a day 25%< multiple times of a day

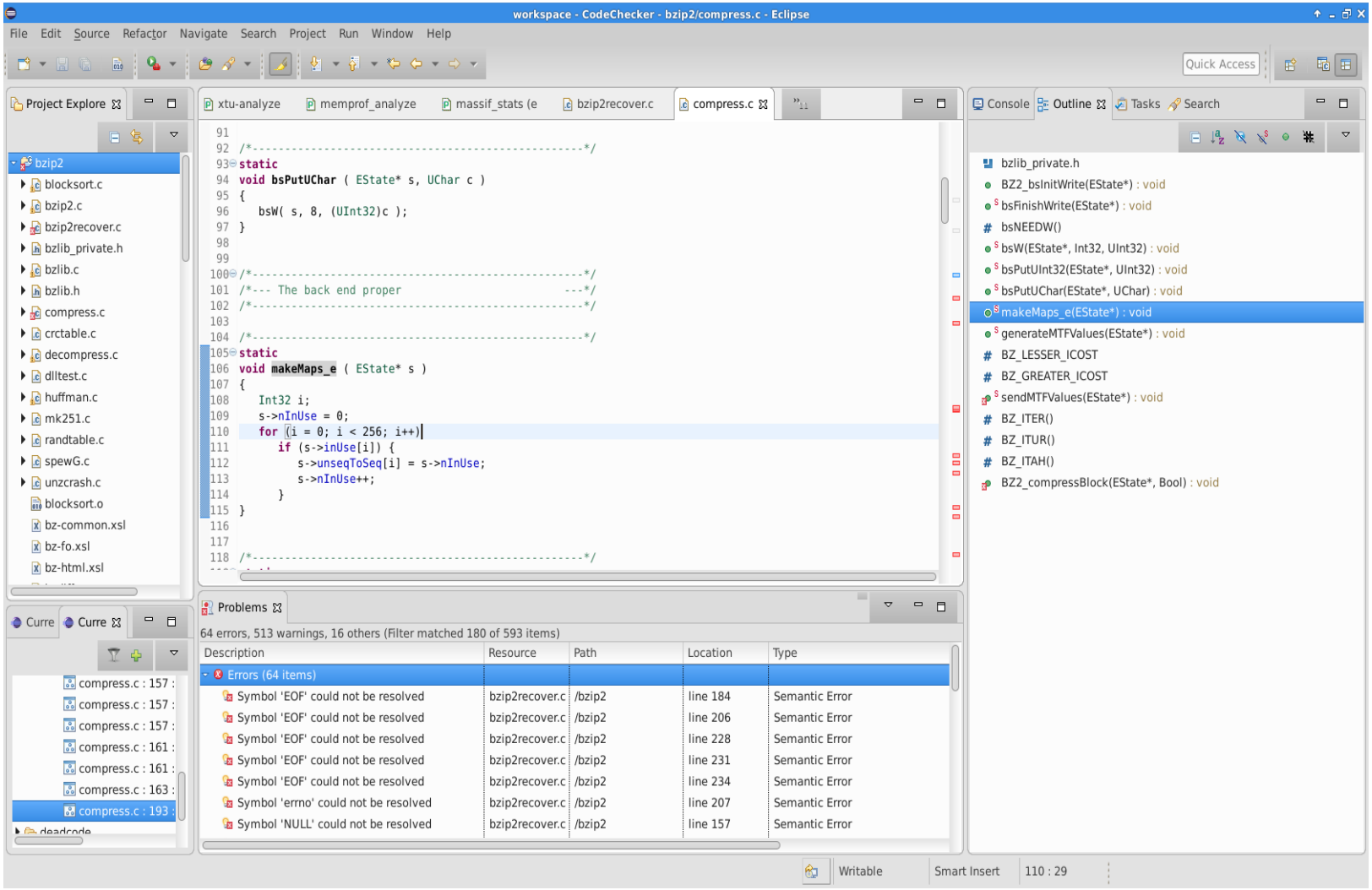
Using tools

Using tools

Using tools



Using tools

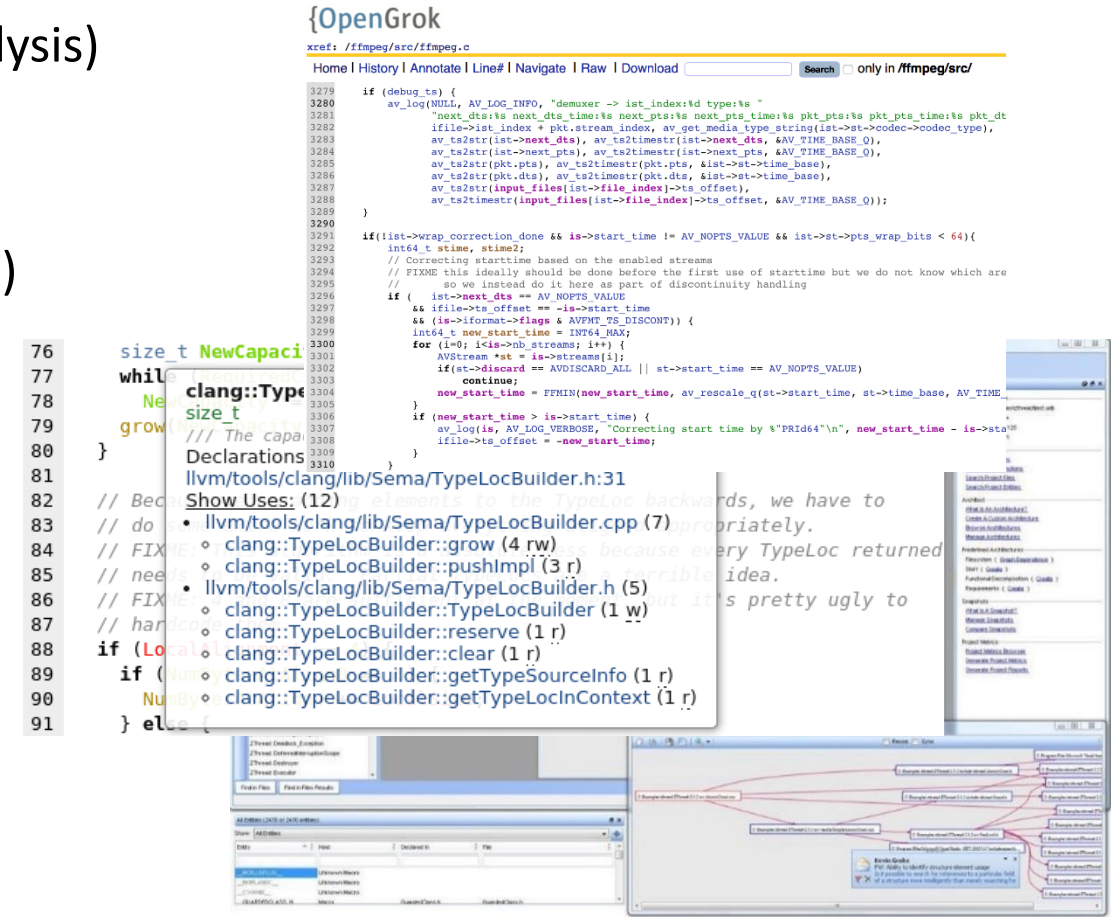

workspace - CodeChecker - bzip2/compress.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explore
bzip2
blocksort.c
bzip2.c
bzip2recover.c
bzlib_private.h
bzlib.c
bzlib.h
compress.c
crctable.c
decompress.c
dlltest.c
huffman.c
mk251.c
randtable.c
spewG.c
unzcrash.c
blocksort.o
bz-common.xsl
bz-fo.xsl
bz-html.xsl
xtu-analyze
memprof_analyze
massif_stats (e
bzip2recover.c
compress.c
/*--- The back end proper ---*/
static
void bsPutUChar (EState* s, UChar c)
{
bsW(s, 8, (UInt32)c);
}
static
void makeMaps_e (EState* s)
{
Int32 i;
s->nInUse = 0;
for (i = 0; i < 256; i++)
if (s->inUse[i]) {
s->unseqToSeq[i] = s->nInUse;
s->nInUse++;
}
}
Console
Outline
Tasks
Search
bzlib_private.h
BZ2_bsInitWrite(EState*) : void
bsFinishWrite(EState*) : void
bsNEEDW()
bsW(EState*, Int32, UInt32) : void
bsPutUInt32(EState*, UInt32) : void
bsPutUChar(EState*, UChar) : void
makeMaps_e(EState*) : void
generateMTFValues(EState*) : void
BZ_LESSER_ICOST
BZ_GREATER_ICOST
sendMTFValues(EState*) : void
BZ_ITER()
BZ_ITUR()
BZ_ITAH()
BZ2_compressBlock(EState*, Bool) : void
Curre
Curre
compress.c : 157 :
compress.c : 157 :
compress.c : 157 :
compress.c : 161 :
compress.c : 161 :
compress.c : 163 :
compress.c : 193 :
deadcode
Problems
64 errors, 513 warnings, 16 others (Filter matched 180 of 593 items)
Description Resource Path Location Type
Errors (64 items)
Symbol 'EOF' could not be resolved bzip2recover.c /bzip2 line 184 Semantic Error
Symbol 'EOF' could not be resolved bzip2recover.c /bzip2 line 206 Semantic Error
Symbol 'EOF' could not be resolved bzip2recover.c /bzip2 line 228 Semantic Error
Symbol 'EOF' could not be resolved bzip2recover.c /bzip2 line 231 Semantic Error
Symbol 'EOF' could not be resolved bzip2recover.c /bzip2 line 234 Semantic Error
Symbol 'errno' could not be resolved bzip2recover.c /bzip2 line 207 Semantic Error
Symbol 'NULL' could not be resolved bzip2recover.c /bzip2 line 157 Semantic Error
Writable
Smart Insert
110 : 29

Comprehension requires specific toolset

Development of code	Understanding code
Writing new code (support: code completion, etc.)	Reading and navigating inside code
Intentions are clear	Intensions are weak
Editing only a few files at the same time	Frequently jumping between different files
Working on the same abstraction level for a while	Jumping between various abstraction levels (Google map of code)
Edit, compile, fix	Visualize

Some existing tools

- Web-based
  - OpenGrok
  - Woboq (deep analysis)
  - …
- Fat-client
  - Understand (+edit)
  - CodeSurfer
  - …
- IDE-based
  - Eclipse
  - NetBeans
  - QtCreator
  - VisualStudio
  - …



Required features

- Deep analysis + build information -> using a real parser
- Fast text based feature location
- Architectural information
- Textual summaries (types, variables, functions, macros)
- Various (interactive) visualizations
- Scalable (>10 million LOC)
- Most actions should be fast (< 1-2 sec)
- Permalinks for communication with fellow developers
- Gathering all available information: code history, metrics, …
- Open, extensible platform

First experimental version: store AST

- AST contains most of the required information
- Natural output of Clang
- Problem: size!
  - 40GB for LLVM project AST dump + indexes, etc… ->100 GB
  - 1:500 ratio between source and CodeCompass DB size
- Not scalable
- Future work:
  - Detecting identical sub-trees (e.g. of headers)
  - NoSQL database?
- Fat client

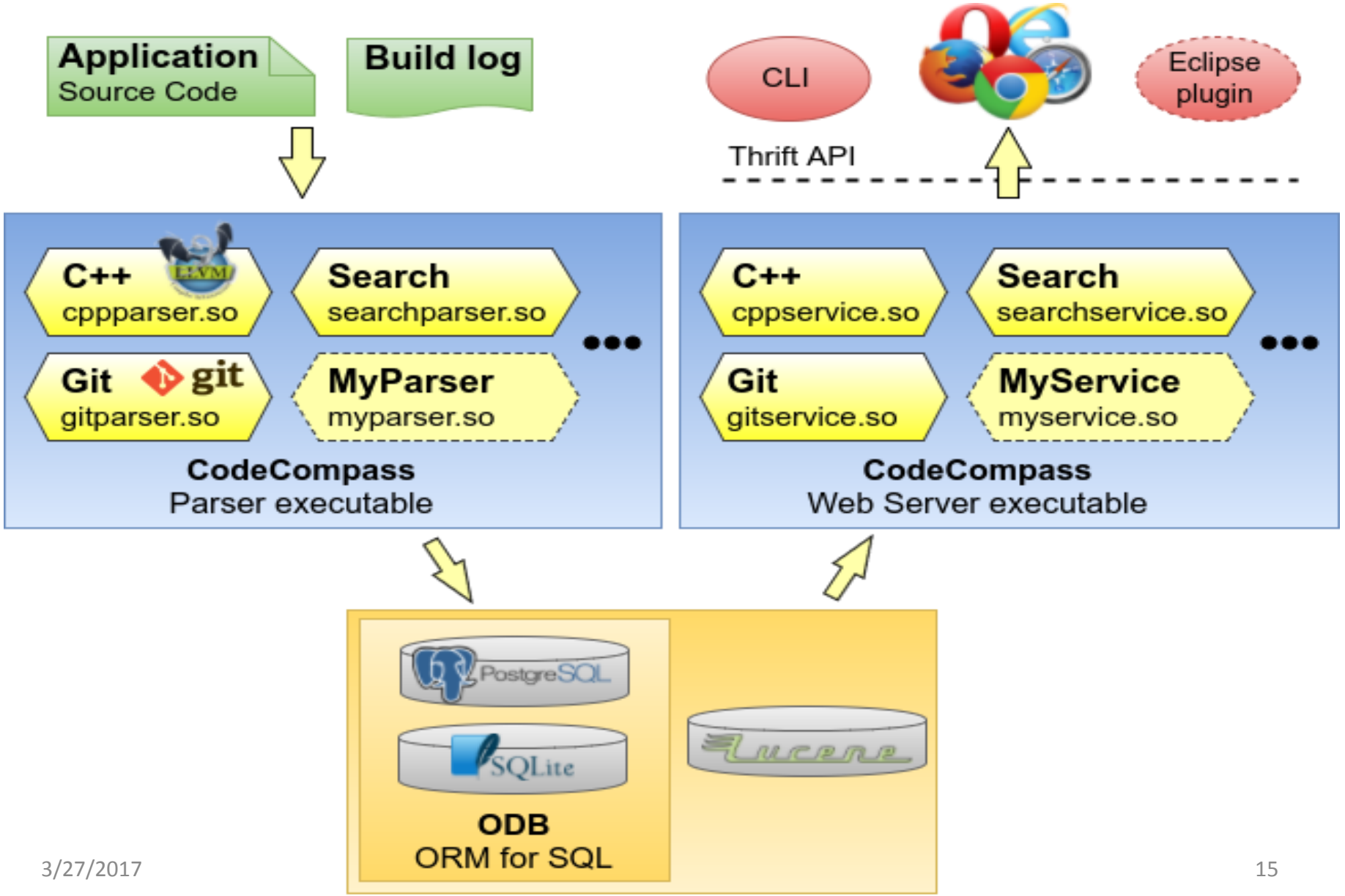
Final approach: Store named entities

- Names: the most natural target of user actions
- We store
  - Class/function/variable declarations, definitions, usage
  - References to names are stored as hash values
  - Source file as it is (keeping original formatting)
  - Build information
- Scalable
  - 1:30-50 ratio between source and CodeCompass DB size
  - Full LLVM CodeCompass DB with indexes 13 GB in postgres
- A few addition was required
  - Assignment, parameter lists: detecting read/write relations of variables
  - Inheritance, pointer indirections, typedefs, etc…
- Web-based client

Performance

	Tiny XML 2.6.2	Xerces 3.1.3	CodeCompass v4	Ericsson TSP product
Source code size [MiB]	1.16	67.28	182	3 344
Search database size [MiB]	0.88	37.93	139	7168
PostgreSQL DB size [MiB]	15	190	2144	7729
Build time [s]	2.73	361	2024	-
CC Parse time [s]	21.98	517	6409	-
Text/definition search [s]	0.4	0.3	0.43	2
C++ get usage of a type [s]	1.4	2	2.3	3.1

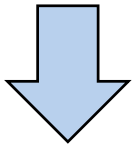
Architecture



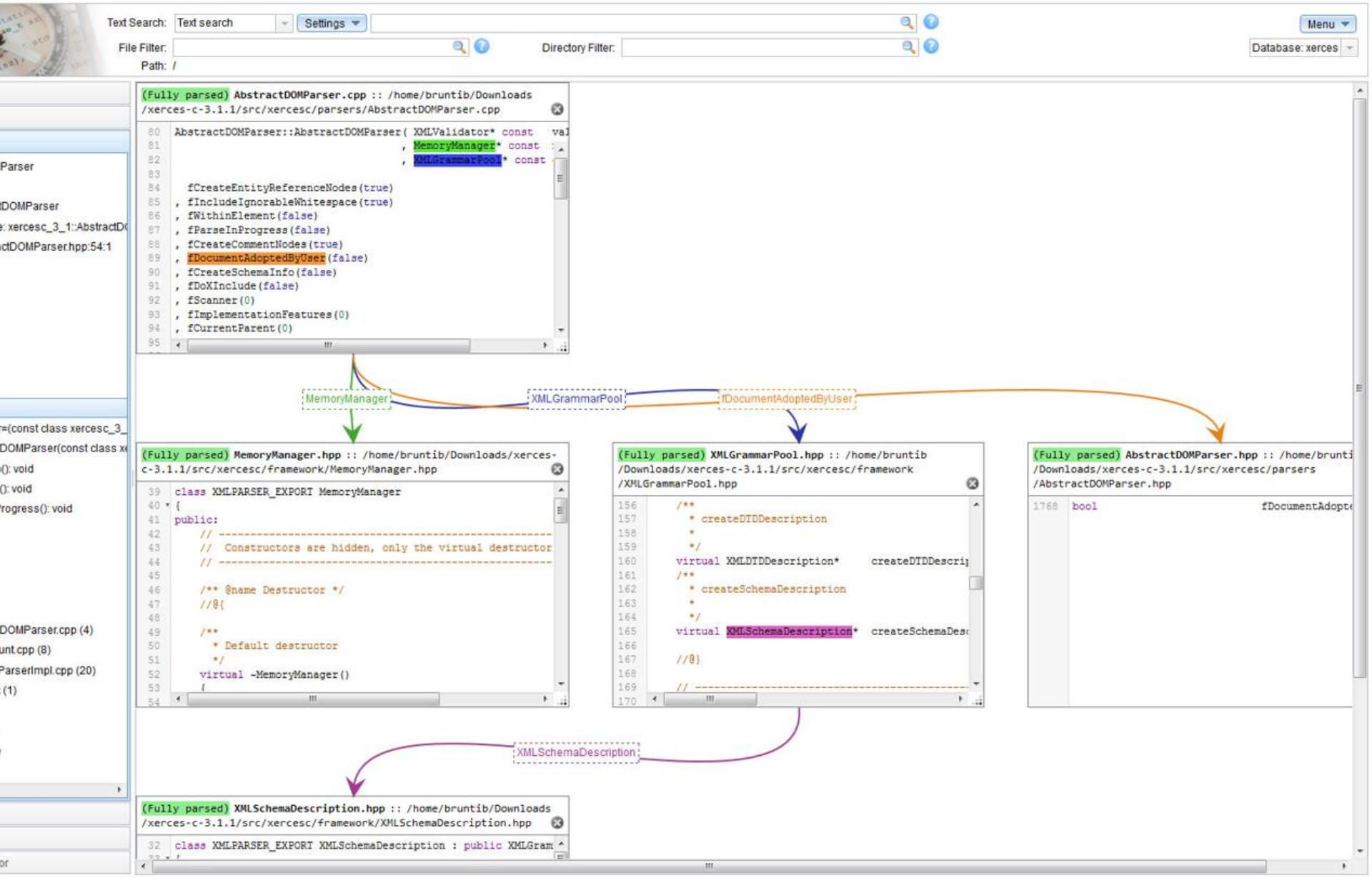
How to use?

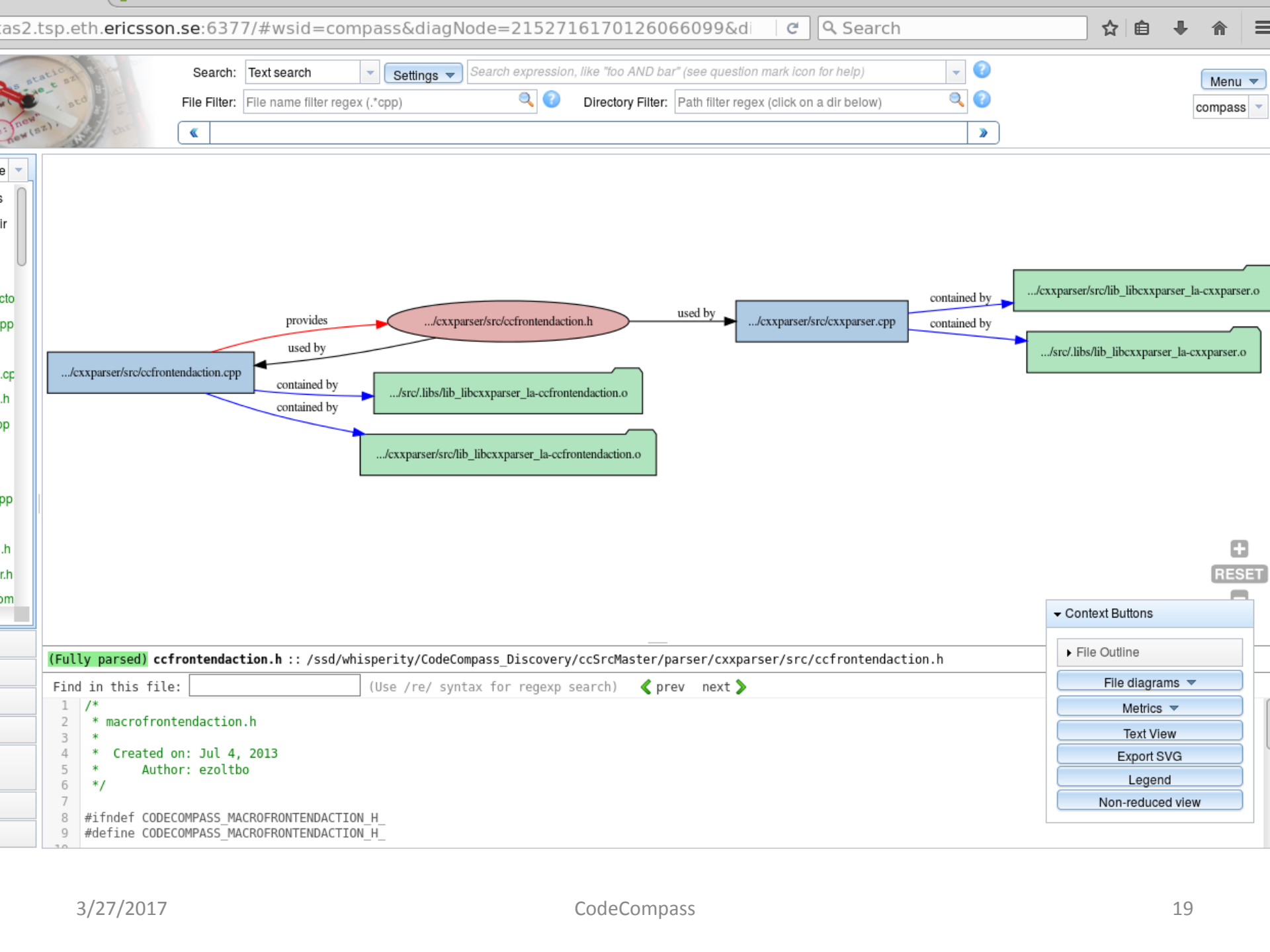
- Fast feature location using text/definition/log search
- Explore the environment of the focus point
  - Info tree
  - Interactive call graphs
  - Virtual functions and function pointers
- Understand the code history
- Understand higher level architecture
- Explore related static analysis results/code metrics

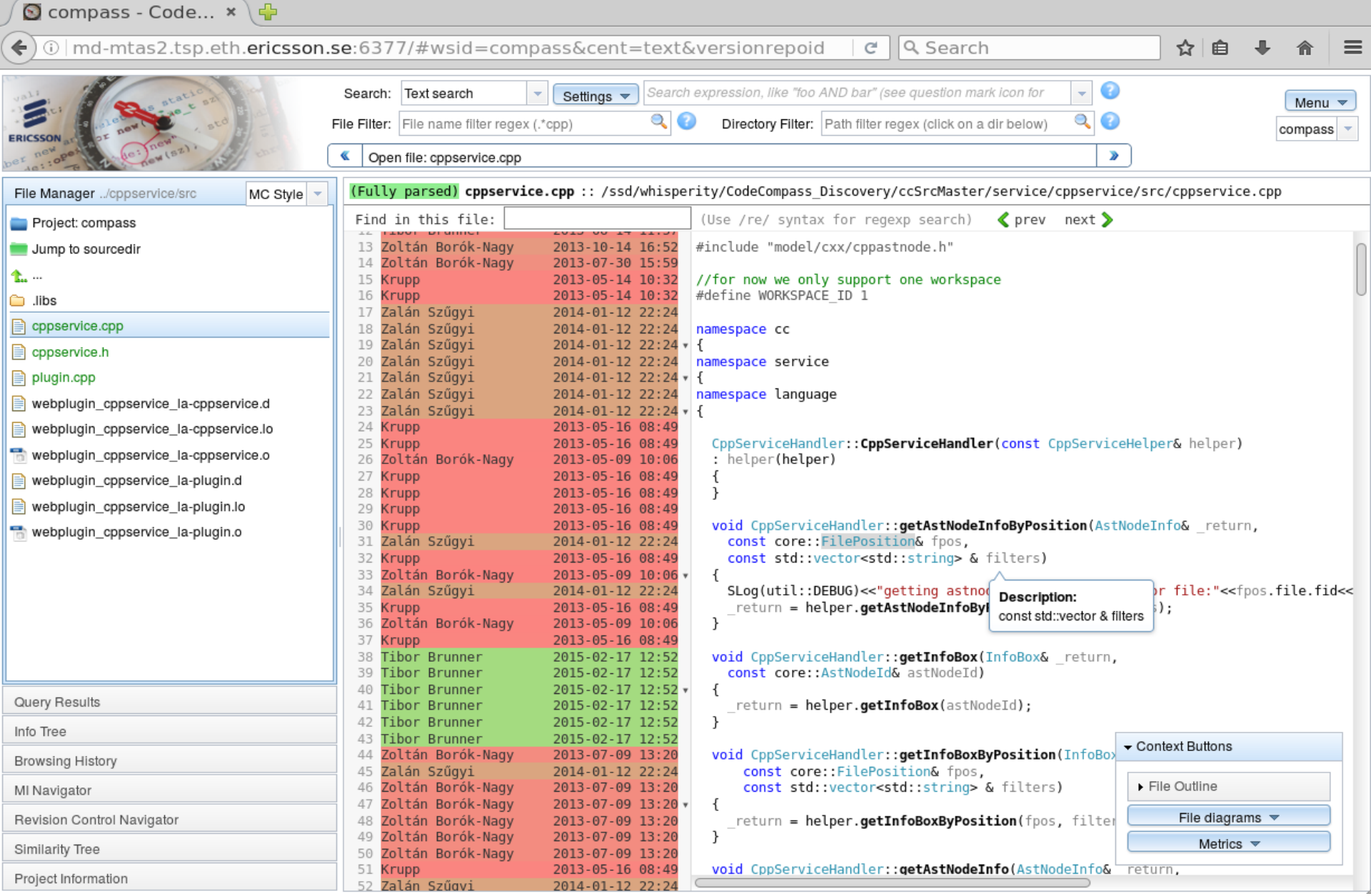
```
DEBUG INFO: TSTHan:          sys_offset=-0.019821, drift_comp=-90.4996, sys_poll=5
```

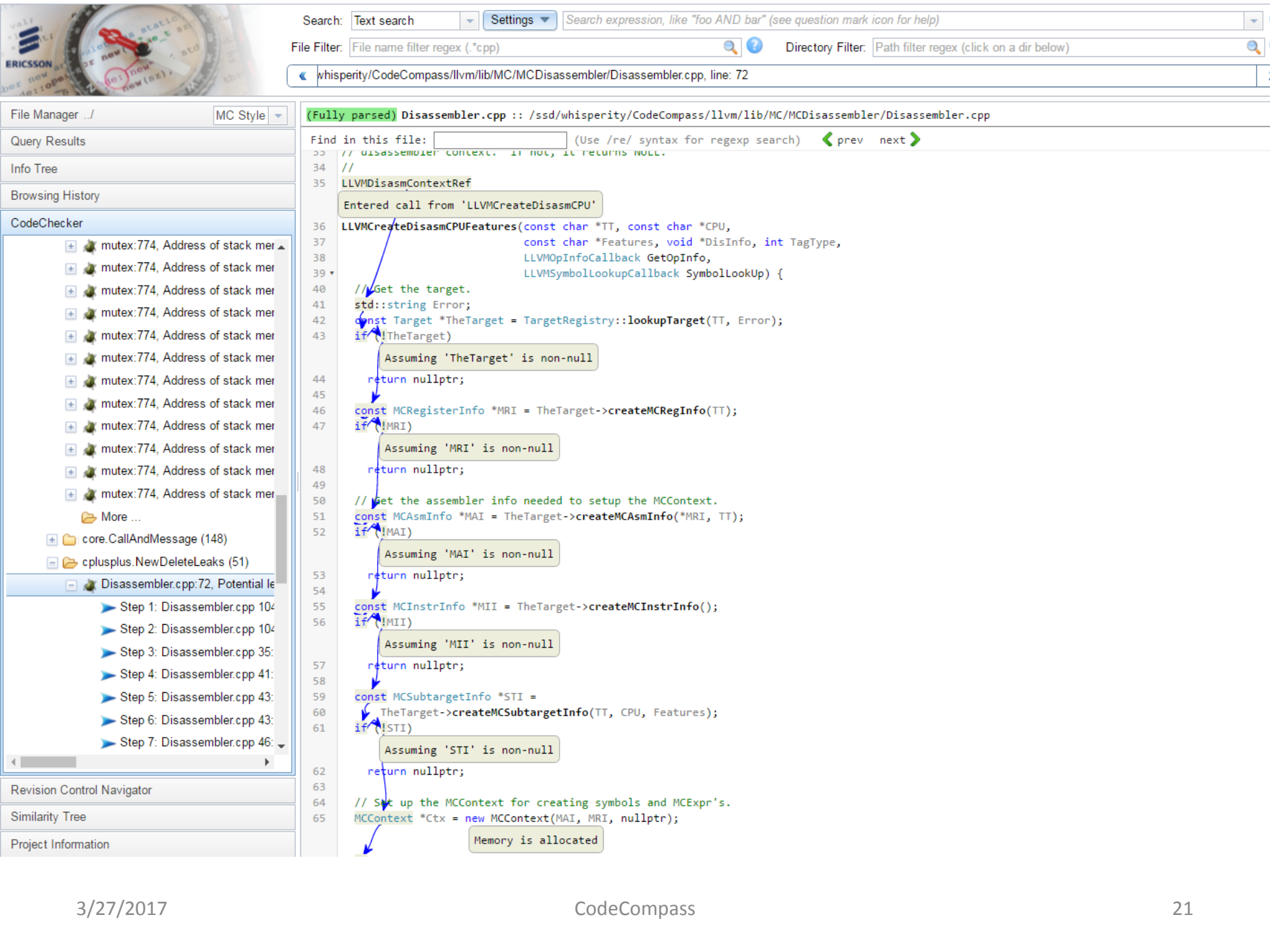


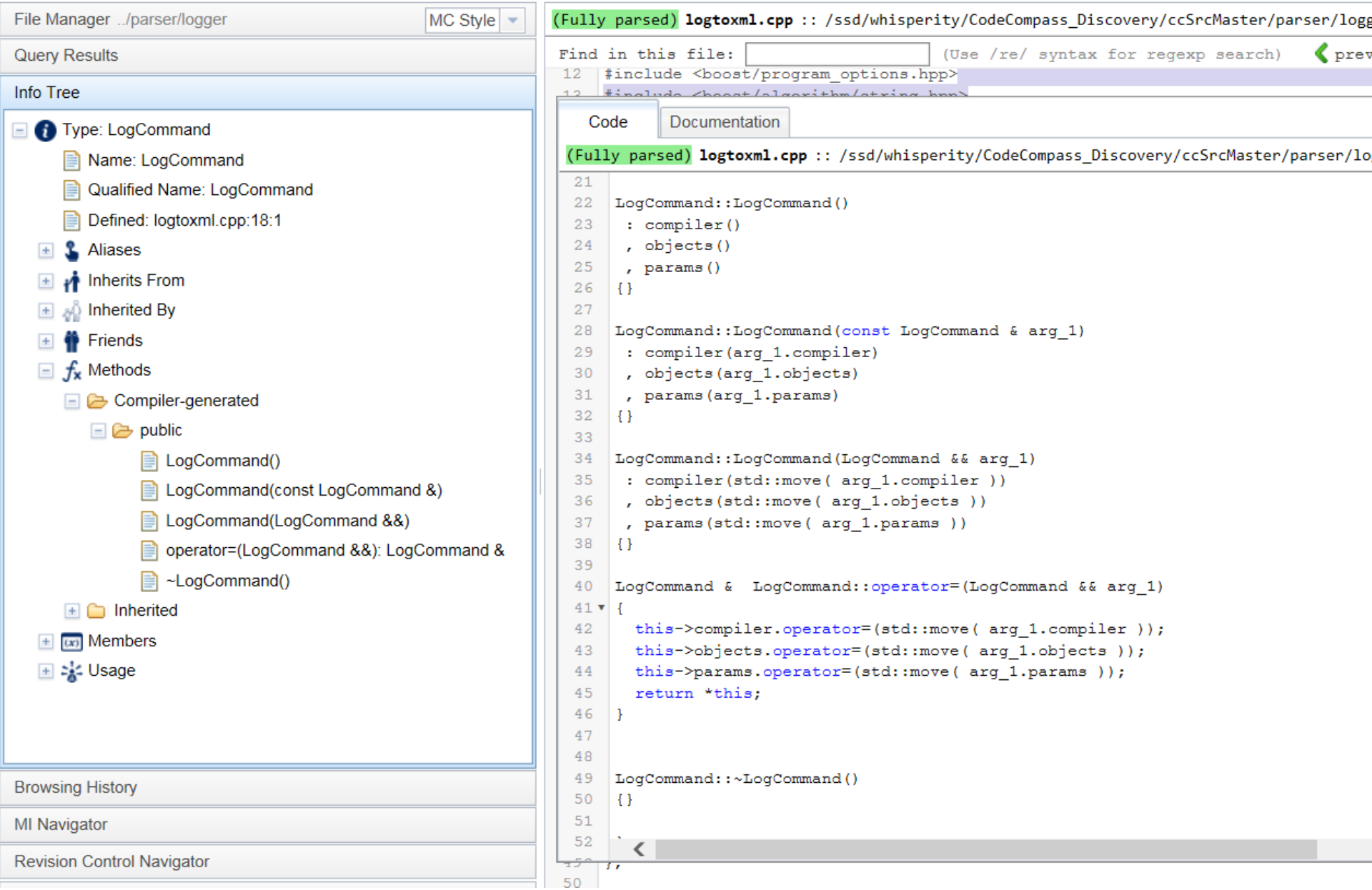
```
818     conprint("    ----------------- NTP hourly stats -----------------\n");
819     ntpq_p();
820     conprint("         sys_offset=%s, drift_comp=%s, sys_poll=%d\n",
821        lfptoa(&sys_offset, 6), fptoa(drift_comp, 4), sys_poll);
822     conprint("    ---------------------------------------------------\n");
823
```


Text Search:
Text search
Settings
Menu
File Filter:
Directory Filter:
Path: /
Database: xerces
(Fully parsed) AbstractDOMParser.cpp :: /home/bruntib/Downloads
/xerces-c-3.1.1/src/xercesc/parsers/AbstractDOMParser.cpp
AbstractDOMParser::AbstractDOMParser(XMLValidator* const
, MemoryManager* const
, XMLGrammarPool* const
fCreateEntityReferenceNodes(true)
, fIncludeIgnorableWhitespace(true)
, fWithinElement(false)
, fParseInProgress(false)
, fCreateCommentNodes(true)
, fDocumentAdoptedByUser(false)
, fCreateSchemaInfo(false)
, fDoXInclude(false)
, fScanner(0)
, fImplementationFeatures(0)
, fCurrentParent(0)
MemoryManager
XMLGrammarPool
fDocumentAdoptedByUser
(Fully parsed) MemoryManager.hpp :: /home/bruntib/Downloads/xerces-
c-3.1.1/src/xercesc/framework/MemoryManager.hpp
class XMLPARSER_EXPORT MemoryManager
{
public:
// Constructors are hidden, only the virtual destructor
/** @name Destructor */
/** Default destructor
virtual ~MemoryManager()
(Fully parsed) XMLGrammarPool.hpp :: /home/bruntib
/Downloads/xerces-c-3.1.1/src/xercesc/framework
/XMLGrammarPool.hpp
/** createDTDDescription
virtual XMLDTDDescription* createDTDDescri
/** createSchemaDescription
virtual XMLSchemaDescription* createSchemaDes
(Fully parsed) AbstractDOMParser.hpp :: /home/brunti
/Downloads/xerces-c-3.1.1/src/xercesc/parsers
/AbstractDOMParser.hpp
bool fDocumentAdopte
XMLSchemaDescription
(Fully parsed) XMLSchemaDescription.hpp :: /home/bruntib/Downloads
/xerces-c-3.1.1/src/xercesc/framework/XMLSchemaDescription.hpp
class XMLPARSER_EXPORT XMLSchemaDescription : public XMLGram


tas2.tsp.eth.ericsson.se:6377/#wsid=compass&diagNode=2152716170126066099&d
Search
Search:
Text search
Settings
Search expression, like "foo AND bar" (see question mark icon for help)
File Filter:
File name filter regex (.*cpp)
Directory Filter:
Path filter regex (click on a dir below)
Menu
compass
provides
used by
.../cxxparser/src/ccfrontendaction.h
used by
.../cxxparser/src/cxxparser.cpp
contained by
contained by
.../cxxparser/src/lib_libcxxparser_la-cxxparser.o
.../src/.libs/lib_libcxxparser_la-cxxparser.o
.../cxxparser/src/ccfrontendaction.cpp
contained by
contained by
.../src/.libs/lib_libcxxparser_la-ccfrontendaction.o
.../cxxparser/src/lib_libcxxparser_la-ccfrontendaction.o
RESET
Context Buttons
File Outline
File diagrams
Metrics
Text View
Export SVG
Legend
Non-reduced view
(Fully parsed) ccfrontendaction.h :: /ssd/whisperity/CodeCompass_Discovery/ccSrcMaster/parser/cxxparser/src/ccfrontendaction.h
Find in this file:
(Use /re/ syntax for regexp search)
prev
next
1 /*
2 * macrofrontendaction.h
3 *
4 * Created on: Jul 4, 2013
5 * Author: ezoltbo
6 */
7
8 #ifndef CODECOMPASS_MACROFRONTENDACTION_H_
9 #define CODECOMPASS_MACROFRONTENDACTION_H_


compass - Code...
md-mtas2.tsp.eth.ericsson.se:6377/#wsid=compass&cent=text&versionrepoid
Search
Search: Text search
Settings
Search expression, like "foo AND bar" (see question mark icon for
File Filter: File name filter regex (.*cpp)
Directory Filter: Path filter regex (click on a dir below)
Menu
compass
Open file: cppservice.cpp
File Manager ../cppservice/src
MC Style
Project: compass
Jump to sourcedir
...
.libs
cppservice.cpp
cppservice.h
plugin.cpp
webplugin_cppservice_la-cppservice.d
webplugin_cppservice_la-cppservice.lo
webplugin_cppservice_la-cppservice.o
webplugin_cppservice_la-plugin.d
webplugin_cppservice_la-plugin.lo
webplugin_cppservice_la-plugin.o
Query Results
Info Tree
Browsing History
MI Navigator
Revision Control Navigator
Similarity Tree
Project Information
(Fully parsed) cppservice.cpp :: /ssd/whisperity/CodeCompass_Discovery/ccSrcMaster/service/cppservice/src/cppservice.cpp
Find in this file:
(Use /re/ syntax for regexp search)
prev
next
13 Zoltán Borók-Nagy 2013-10-14 16:52
14 Zoltán Borók-Nagy 2013-07-30 15:59
15 Krupp 2013-05-14 10:32
16 Krupp 2013-05-14 10:32
17 Zalán Szűgyi 2014-01-12 22:24
18 Zalán Szűgyi 2014-01-12 22:24
19 Zalán Szűgyi 2014-01-12 22:24
20 Zalán Szűgyi 2014-01-12 22:24
21 Zalán Szűgyi 2014-01-12 22:24
22 Zalán Szűgyi 2014-01-12 22:24
23 Zalán Szűgyi 2014-01-12 22:24
24 Krupp 2013-05-16 08:49
25 Krupp 2013-05-16 08:49
26 Zoltán Borók-Nagy 2013-05-09 10:06
27 Krupp 2013-05-16 08:49
28 Krupp 2013-05-16 08:49
29 Krupp 2013-05-16 08:49
30 Krupp 2013-05-16 08:49
31 Zalán Szűgyi 2014-01-12 22:24
32 Krupp 2013-05-16 08:49
33 Zoltán Borók-Nagy 2013-05-09 10:06
34 Zalán Szűgyi 2014-01-12 22:24
35 Krupp 2013-05-16 08:49
36 Zoltán Borók-Nagy 2013-05-09 10:06
37 Krupp 2013-05-16 08:49
38 Tibor Brunner 2015-02-17 12:52
39 Tibor Brunner 2015-02-17 12:52
40 Tibor Brunner 2015-02-17 12:52
41 Tibor Brunner 2015-02-17 12:52
42 Tibor Brunner 2015-02-17 12:52
43 Tibor Brunner 2015-02-17 12:52
44 Zoltán Borók-Nagy 2013-07-09 13:20
45 Zalán Szűgyi 2014-01-12 22:24
46 Zoltán Borók-Nagy 2013-07-09 13:20
47 Zoltán Borók-Nagy 2013-07-09 13:20
48 Zoltán Borók-Nagy 2013-07-09 13:20
49 Zoltán Borók-Nagy 2013-07-09 13:20
50 Zoltán Borók-Nagy 2013-07-09 13:20
51 Krupp 2013-05-16 08:49
52 Zalán Szűgyi 2014-01-12 22:24
#include "model/cxx/cppastnode.h"
//for now we only support one workspace
#define WORKSPACE_ID 1
namespace cc
{
namespace service
{
namespace language
{
CppServiceHandler::CppServiceHandler(const CppServiceHelper& helper)
: helper(helper)
{
}
void CppServiceHandler::getAstNodeInfoByPosition(AstNodeInfo& _return,
const core::FilePosition& fpos,
const std::vector<std::string> & filters)
{
SLog(util::DEBUG)<<"getting astnod
_return = helper.getAstNodeInfoBy
}
Description:
const std::vector & filters
r file:"<<fpos.file.fid<<
);
void CppServiceHandler::getInfoBox(InfoBox& _return,
const core::AstNodeId& astNodeId)
{
_return = helper.getInfoBox(astNodeId);
}
void CppServiceHandler::getInfoBoxByPosition(InfoBox
const core::FilePosition& fpos,
const std::vector<std::string> & filters)
{
_return = helper.getInfoBoxByPosition(fpos, filter
}
void CppServiceHandler::getAstNodeInfo(AstNodeInfo& _return,
Context Buttons
File Outline
File diagrams
Metrics

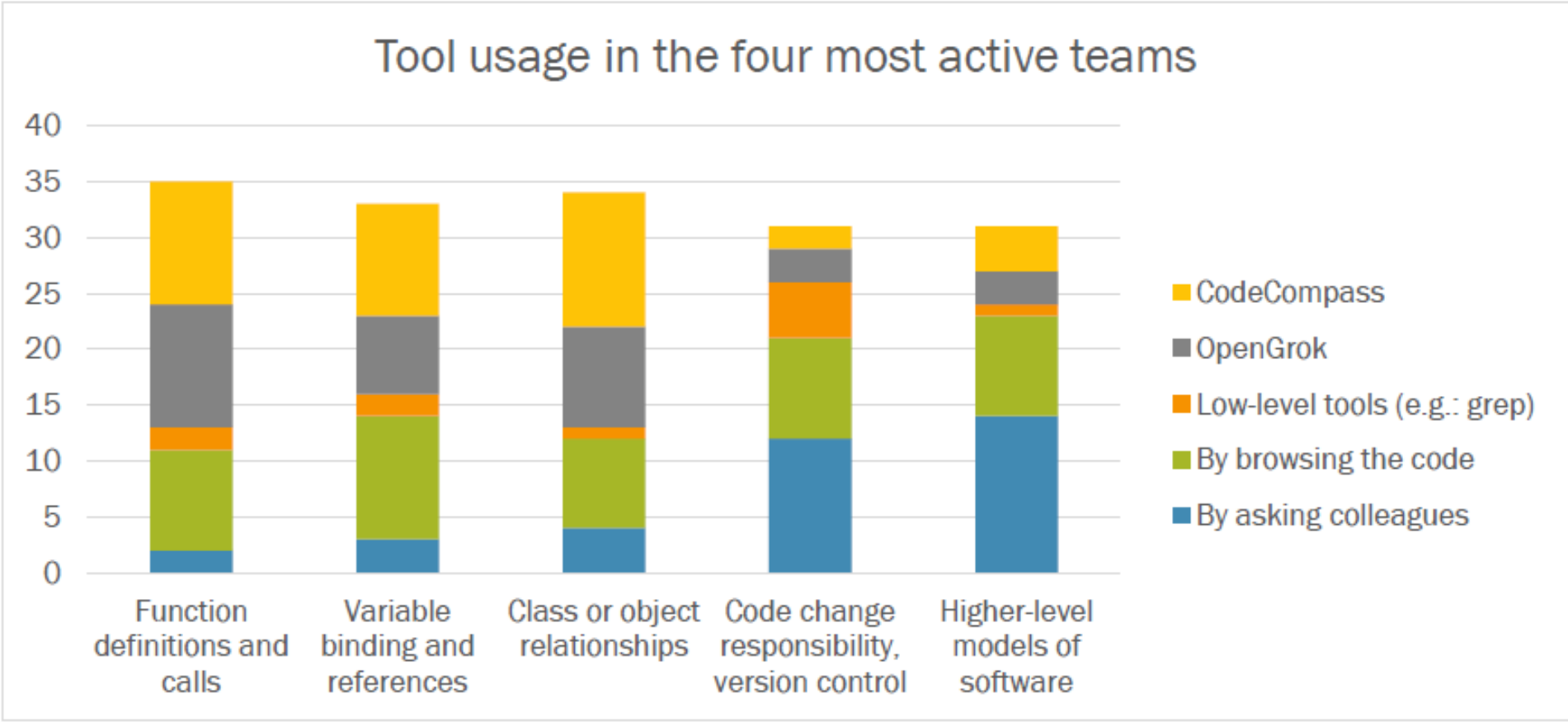

Search: Text search
Settings
Search expression, like "foo AND bar" (see question mark icon for help)
File Filter: File name filter regex (.*cpp)
Directory Filter: Path filter regex (click on a dir below)
whisperity/CodeCompass/llvm/lib/MC/MCDisassembler/Disassembler.cpp, line: 72
File Manager ../
MC Style
Query Results
Info Tree
Browsing History
CodeChecker
mutex:774, Address of stack mer
More ...
core.CallAndMessage (148)
cplusplus.NewDeleteLeaks (51)
Disassembler.cpp:72, Potential le
Step 1: Disassembler.cpp 104
Step 2: Disassembler.cpp 104
Step 3: Disassembler.cpp 35:
Step 4: Disassembler.cpp 41:
Step 5: Disassembler.cpp 43:
Step 6: Disassembler.cpp 43:
Step 7: Disassembler.cpp 46:
Revision Control Navigator
Similarity Tree
Project Information
(Fully parsed) Disassembler.cpp :: /ssd/whisperity/CodeCompass/llvm/lib/MC/MCDisassembler/Disassembler.cpp
Find in this file:
(Use /re/ syntax for regexp search)
prev
next
// disassembler context. If not, it returns NULL.
//
LLVMDisasmContextRef
Entered call from 'LLVMCreateDisasmCPU'
LLVMCreateDisasmCPUFeatures(const char *TT, const char *CPU,
const char *Features, void *DisInfo, int TagType,
LLVMOpInfoCallback GetOpInfo,
LLVMSymbolLookupCallback SymbolLookUp) {
// Get the target.
std::string Error;
const Target *TheTarget = TargetRegistry::lookupTarget(TT, Error);
if (!TheTarget)
Assuming 'TheTarget' is non-null
return nullptr;
const MCRegisterInfo *MRI = TheTarget->createMCRegInfo(TT);
if (!MRI)
Assuming 'MRI' is non-null
return nullptr;
// Get the assembler info needed to setup the MCContext.
const MCAsmInfo *MAI = TheTarget->createMCAsmInfo(*MRI, TT);
if (!MAI)
Assuming 'MAI' is non-null
return nullptr;
const MCInstrInfo *MII = TheTarget->createMCInstrInfo();
if (!MII)
Assuming 'MII' is non-null
return nullptr;
const MCSubtargetInfo *STI =
TheTarget->createMCSubtargetInfo(TT, CPU, Features);
if (!STI)
Assuming 'STI' is non-null
return nullptr;
// Set up the MCContext for creating symbols and MCExpr's.
MCContext *Ctx = new MCContext(MAI, MRI, nullptr);
Memory is allocated


File Manager ../parser/logger
MC Style
Query Results
Info Tree
Type: LogCommand
Name: LogCommand
Qualified Name: LogCommand
Defined: logtoxml.cpp:18:1
Aliases
Inherits From
Inherited By
Friends
Methods
Compiler-generated
public
LogCommand()
LogCommand(const LogCommand &)
LogCommand(LogCommand &&)
operator=(LogCommand &&): LogCommand &
~LogCommand()
Inherited
Members
Usage
Browsing History
MI Navigator
Revision Control Navigator
(Fully parsed) logtoxml.cpp :: /ssd/whisperity/CodeCompass_Discovery/ccSrcMaster/parser/logg
Find in this file:
(Use /re/ syntax for regexp search)
prev
12 #include <boost/program_options.hpp>
Code
Documentation
(Fully parsed) logtoxml.cpp :: /ssd/whisperity/CodeCompass_Discovery/ccSrcMaster/parser/lo
21
22 LogCommand::LogCommand()
23 : compiler()
24 , objects()
25 , params()
26 {}
27
28 LogCommand::LogCommand(const LogCommand & arg_1)
29 : compiler(arg_1.compiler)
30 , objects(arg_1.objects)
31 , params(arg_1.params)
32 {}
33
34 LogCommand::LogCommand(LogCommand && arg_1)
35 : compiler(std::move(arg_1.compiler))
36 , objects(std::move(arg_1.objects))
37 , params(std::move(arg_1.params))
38 {}
39
40 LogCommand & LogCommand::operator=(LogCommand && arg_1)
41 {
42 this->compiler.operator=(std::move(arg_1.compiler));
43 this->objects.operator=(std::move(arg_1.objects));
44 this->params.operator=(std::move(arg_1.params));
45 return *this;
46 }
47
48
49 LogCommand::~LogCommand()
50 {}
51
52

Experiences with CodeCompass

- Open source since summer 2016
- Mainly used inside Ericsson and in University
- Replacing/extending OpenGrok
- Voluntary-based: No policy to enforce using CodeCompass
- ~15 million LOC parsed inside Ericsson
- ~300 users
- Frequently used investigate CodeChecker results
- … and by architects to get a system level view

Experiences with CodeCompass



Future plans

- Incremental parsers: from “Snapshot” view to editable
  - Pointer analysis
  - Reparse: source + build info -> rebuild AST on demand
- Complex query language
- User specific information
  - Review notes, reminders, comprehension map
  - Personal “Comprehension map” (incl. internal links)
- Ideal for starting a Clang-based server implementing C/C++ LSP (Language Server Protocol), like ClangD
- Feel free to contribute
  - New language parsers
  - New GUI functionality
- Language Server Protocol (LSP) interface

Summary

- Scalable (up to 10 million LOC)
- Most actions are completed (< 1-2sec)
- Textual summaries (types, functions, variables, macros)
- Various (interactive) visualizations on the code
- Architectural information (based on build info)
- GIT history
- Permalinks to communicate with other developers
- CodeChecker integration to show Clang SA results
- Java, Python support (less mature)
- Easy to extend