

C++ Meetup Madrid

Febrero 2019

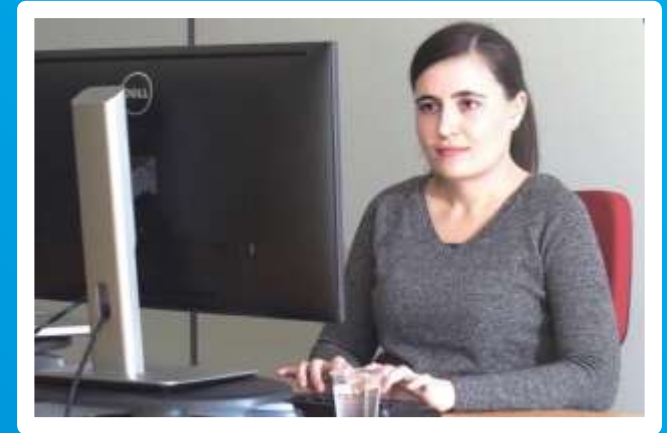


@plasticscm @gmasterscm @semanticmerge



Violeta Sánchez
Software Engineer
[@vio_woman](#)

Míryam Gómez
Software Engineer
[@miryamgsm](#)





Objetivo de la presentación

- **Introducción: empresa y productos.**
- **Motivación del desarrollo de un parser de C++.**
- **Parser de C++ basado en Eclipse CDT.**
- **Aplicaciones prácticas:**
 - Herramienta de merge: semanticmerge
 - Control de versiones: plastic y gmaster
 - Bot que resuelve conflictos de PR sin intervención manual: mergedroid





Código Software

**Quiénes somos y
qué productos
hacemos**



@plasticscm @gmasterscm @semanticmerge

Código Software

- Arrancamos en 2005 (llevamos en control de versiones 14 años!!)
- Situados en el Parque Tecnológico de Boecillo (Valladolid)



<https://www.plasticscm.com/company/team.html>



@plasticscm @gmasterscm @semanticmerge

Cientes en más de 20 países

Financial Sector



IT & Embedded SW



Automobile & Transportation



Defense & Gov.



Games & 3D Design



Medical Devices



Utilities, Logistics & Misc



Productos

Comercial



Distributed Version Control System

Essential tool for software development



Semantic Merge

Language aware diff & merge tool for any VCS

In beta
ready to launch



gmaster

A Git Client Tool with all UI & Semantic Technology (Beta)

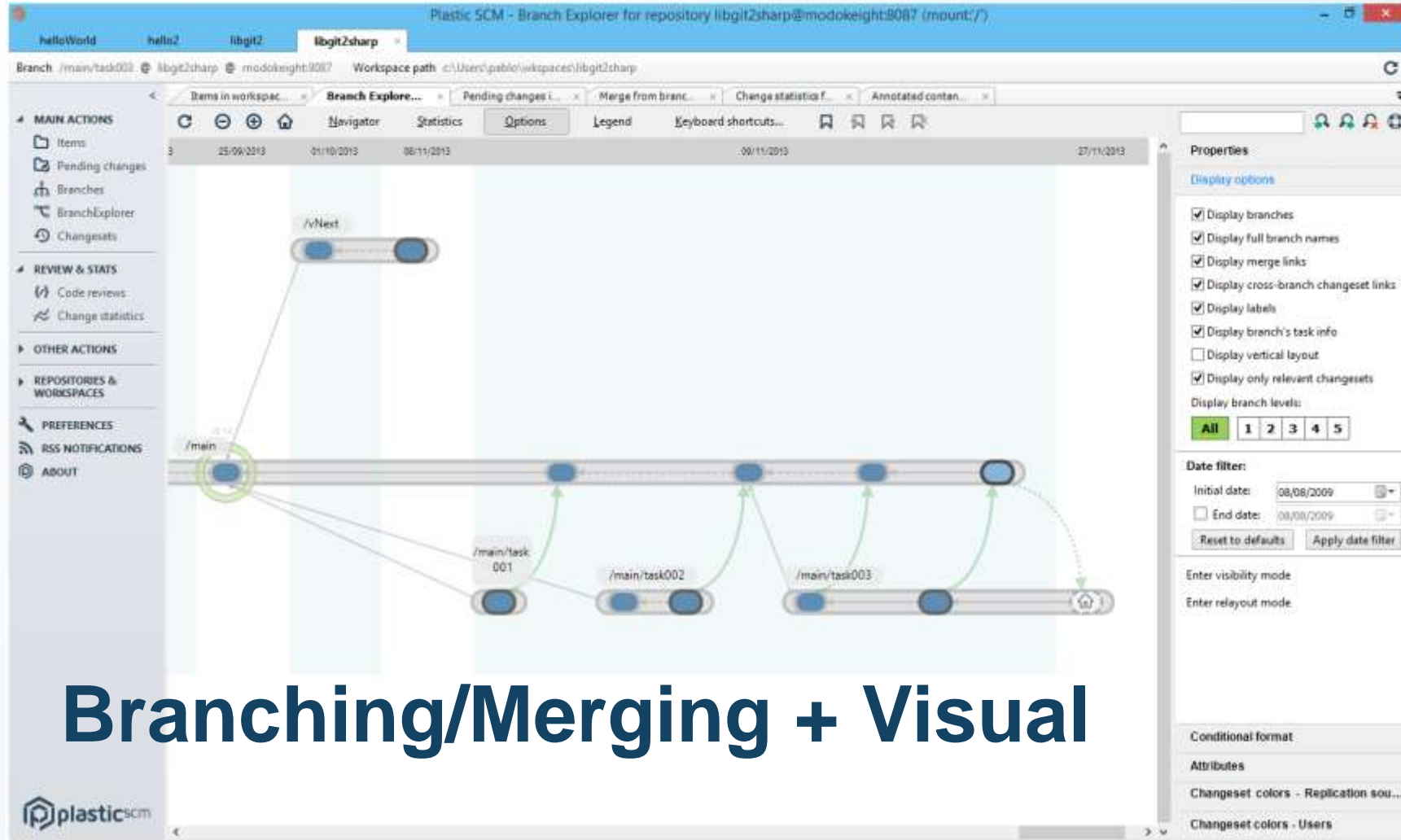


Mergedroid

A server side semantic merge engine tool (Beta)

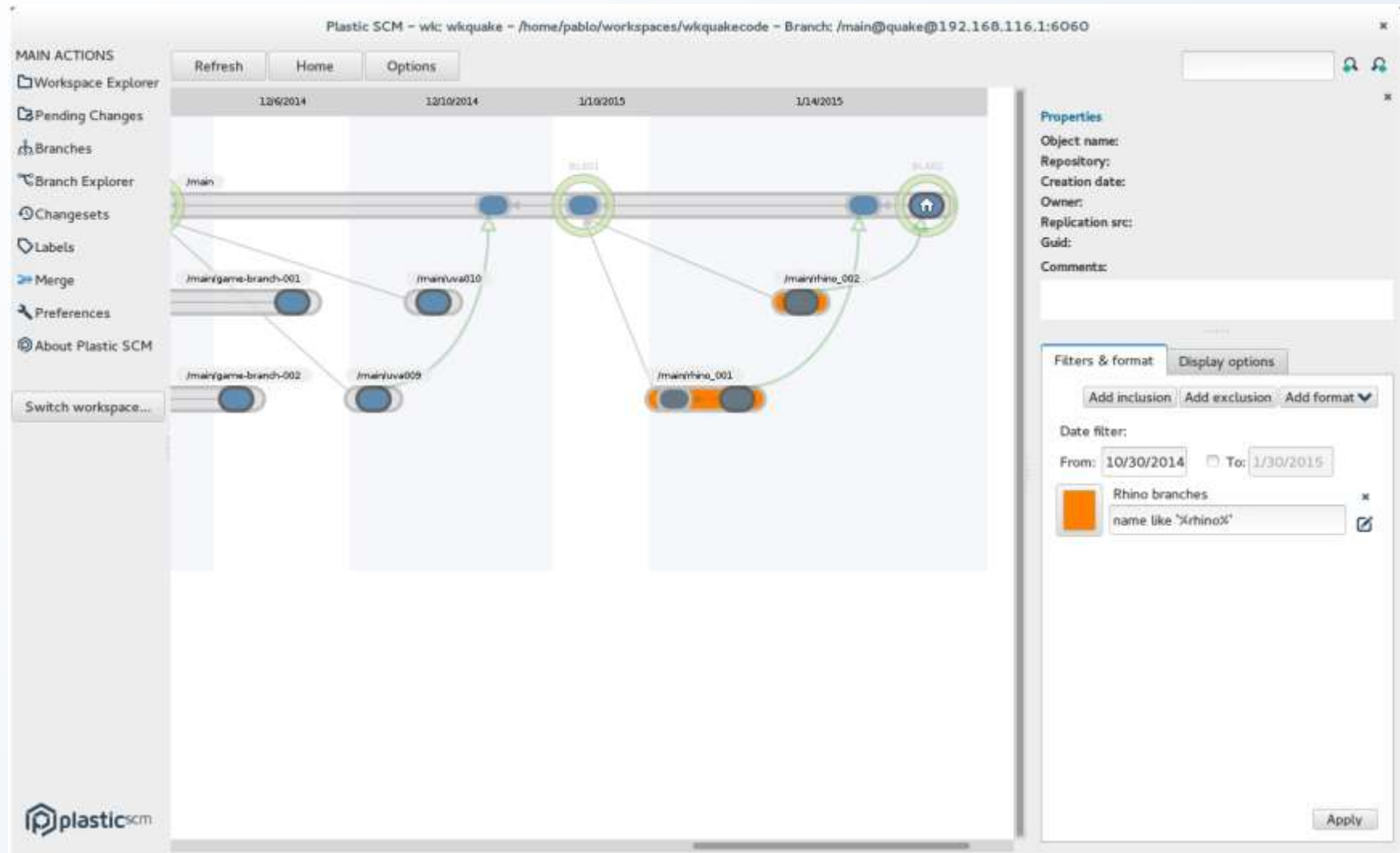


@plasticscm @gmasterscm @semanticmerge

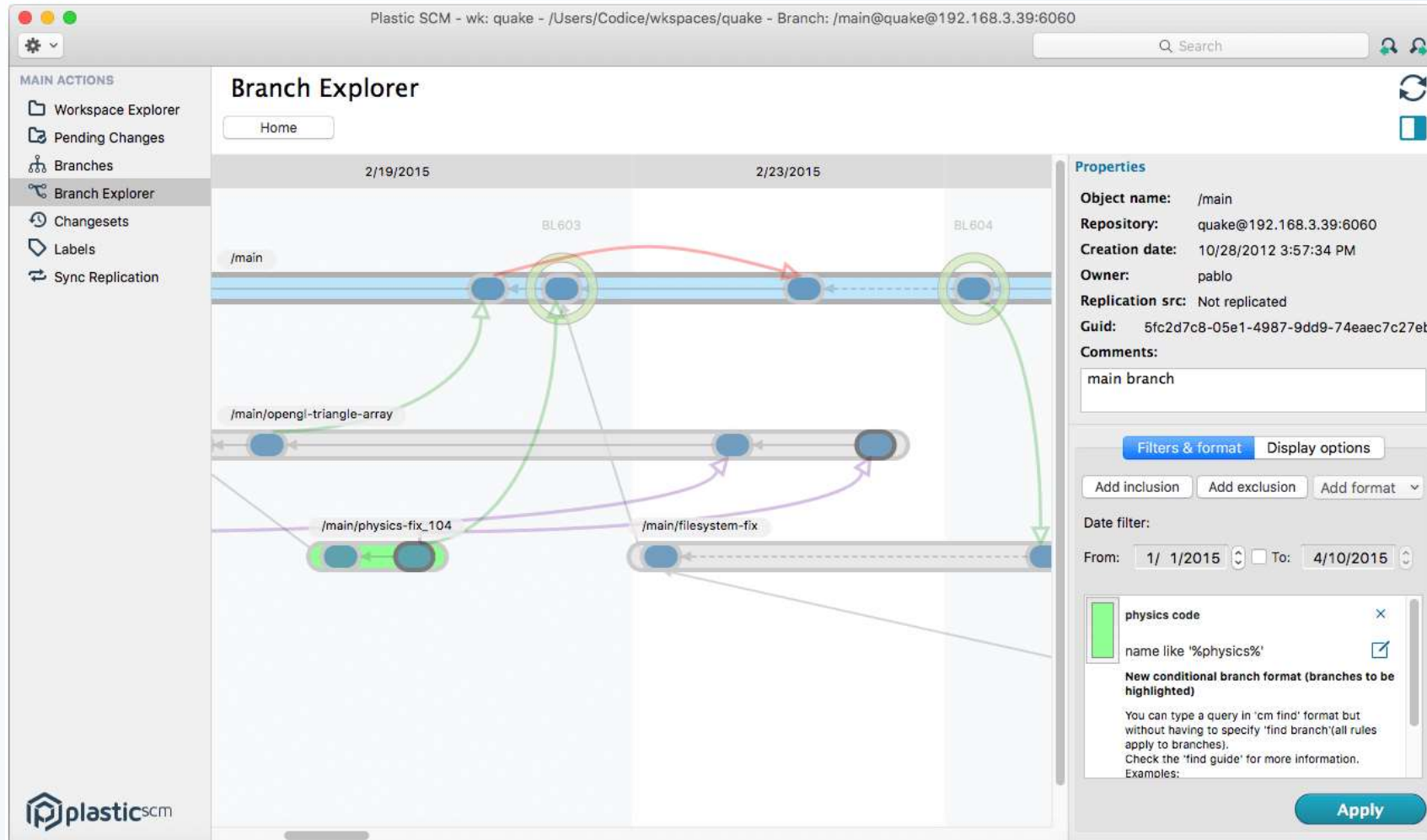


Branching/Merging + Visual

Linux GUI







OS X GUI



Muchos en vídeo juegos


<https://www.plasticscm.com/games/success-stories/index.html>

	<p>+350 DEVELOPERS EPISODIC GAMES</p> <p>Game of Thrones Batman Walking Dead Minecraft Story Mode</p> <p>+4TB repos Artists GUI Network speed Tools team supp. Scalability</p> <p>READ THE TESTIMONIAL</p>		<p>+100 DEVELOPERS</p> <p>Meta 2 Virtual Reality hardware and SDK</p> <p>Unity GUI - Branch Expl. Artists GUI Huge Repos Big files Performance</p> <p>READ THE TESTIMONIAL</p>
	<p>40 DEVELOPERS (FULLY DISTRIBUTED)</p> <p>Rust - Survival game</p> <p>Unity & Unreal GUI - Branch Expl. Artists GUI Huge Repos Big files Performance</p> <p>READ THE TESTIMONIAL</p>		<p>30 DEVELOPERS</p> <p>Open world survival game Subnautica PC and Xbox</p> <p>Big files Huge Repos Branching & Merg. GUI - Branch Expl.</p> <p>READ THE TESTIMONIAL</p>



@plasticscm @gmasterscm @semanticmerge

Gluon (Windows – artistas games)

 Workspace: quakecode

Explore workspace Checkin changes Changesets

Refresh


Configure

Update workspace

Search files...

Details >

Item	Status	Size	Date modified	Changeset	Last edi
[-] c:\Users\pablo\workspaces\test...	Cont...		2/12/2016 14:02:45	649	pablo
[-] art	Cont...		7/7/2015 10:31:13	566	pablo
[-] ironman	Cont...		3/12/2015 16:40:02	511	pablo
[-] covers	Cont...		3/7/2015 19:57:40	506	pablo
[-] volume01	Cont...		3/12/2015 16:40:02	511	pablo
[-] volume03	Cont...		6/19/2014 17:33:33	390	pablo
[-] IronMan.jpg	Cont...	74.66 KB	12/4/2014 19:23:37	466	pablo
[-] warmachinetop.jpg	Cont...	44.88 KB	6/19/2014 17:19:50	387	pablo
[-] screenshots	Cont...		7/7/2015 10:31:13	566	pablo
[-] armory.jpg	Cont...	435.98...	4/27/2015 17:56:23	528	pablo
[-] old_car.png	Cont...	1009.8...	3/7/2015 19:57:40	506	pablo
[-] code	Cont...		2/12/2016 14:02:45	649	pablo
[-] common	Cont...		7/19/2015 19:37:09	579	pablo
[-] ConsoleApplication	Cont...		8/7/2015 11:22:37	597	pablo
[-] HelloWorld	Cont...		2/11/2016 23:06:31	647	pablo
[-] lcc	Cont...		11/21/2014 19:51:13	460	pablo
[-] library	Cont...		6/26/2015 18:23:04	548	pablo



Changeset	Last edited by	Creation date
381	pablo	6/19/2014 17:16
380	pablo	6/19/2014 17:16
379	pablo	6/19/2014 17:15
378	pablo	6/19/2014 17:15
377	pablo	6/19/2014 17:14
376	pablo	6/19/2014 17:13

You are browsing branch /main/scm17542@quake@localhost:6060

Switch workspace...



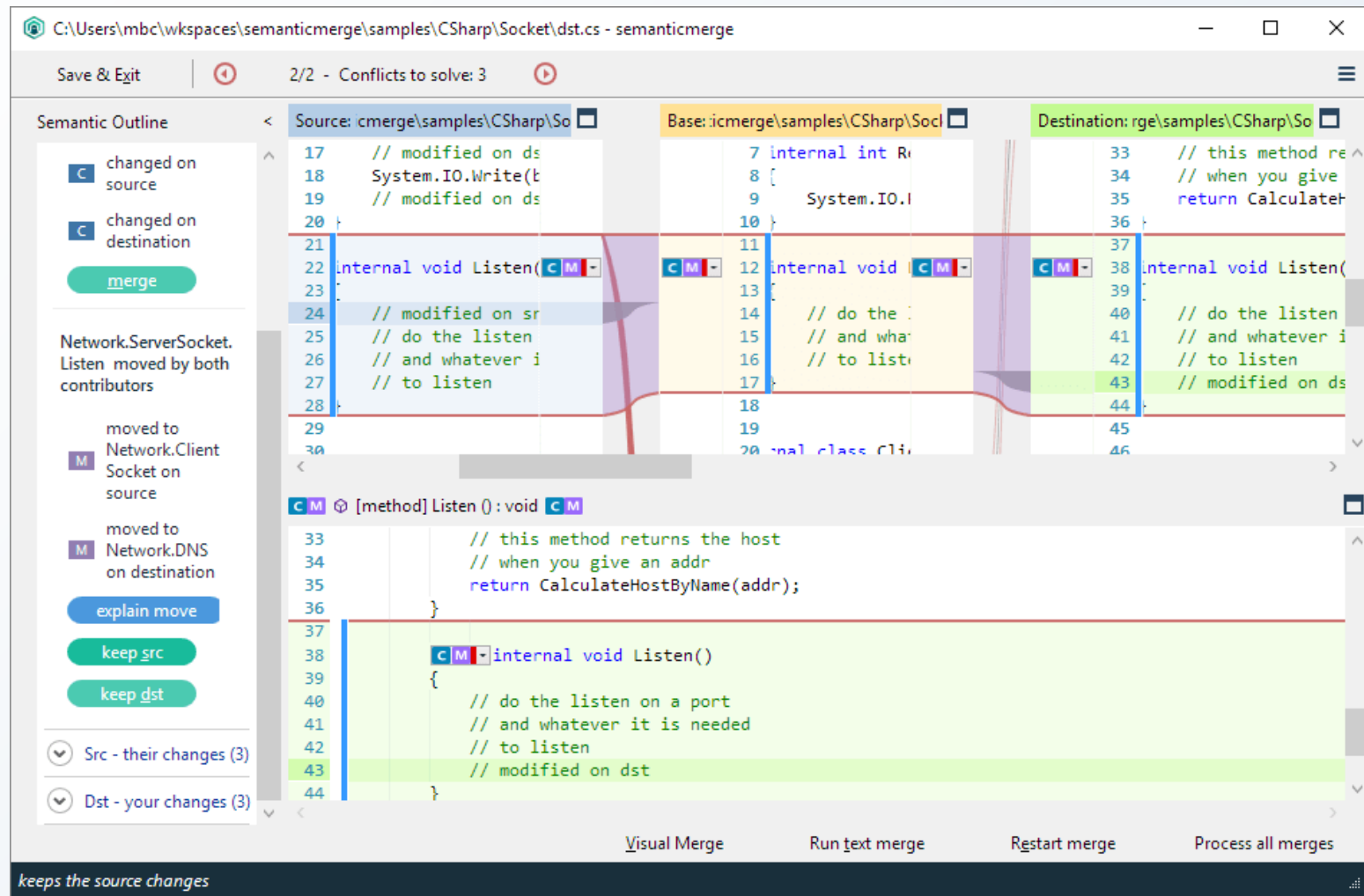


@plasticscm @gmasterscm @semanticmerge



Semantic Merge

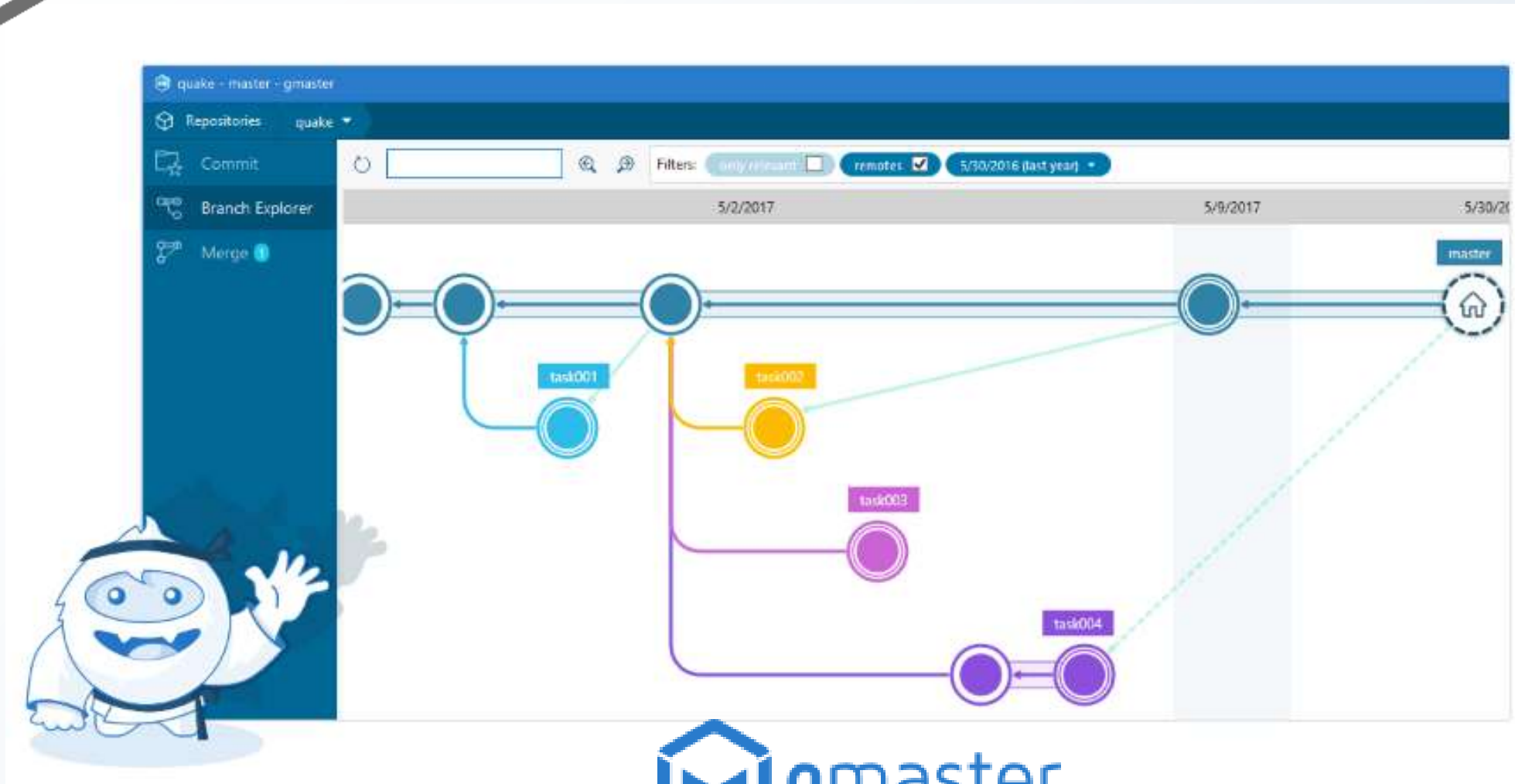
Language aware diff & merge tool for any VCS





gmaster

A Git Client Tool with all UI & Semantic Technology (Beta)



@plasticscm @gmasterscm @semanticmerge



Desarrollo de un parser de C++



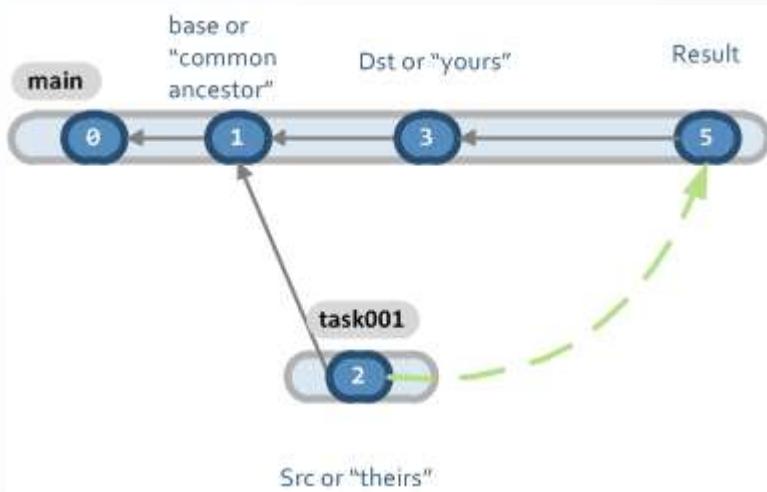
@plasticscm @gmasterscm @semanticmerge

El reto a conseguir

- Principales características de PlasticSCM:
 - Gestión de ramas: patrón rama por tarea.
 - Sistema de merge: a nivel de estructura de directorios, ficheros y estructura dentro del fichero.
- **Reto:** crear una buena herramienta de merge.
- **Objetivo:** hacer el proceso de refactorización sencillo.
- Hasta ahora todas las herramientas de merge se basan en texto, reconocen línea a línea sin entender el código.

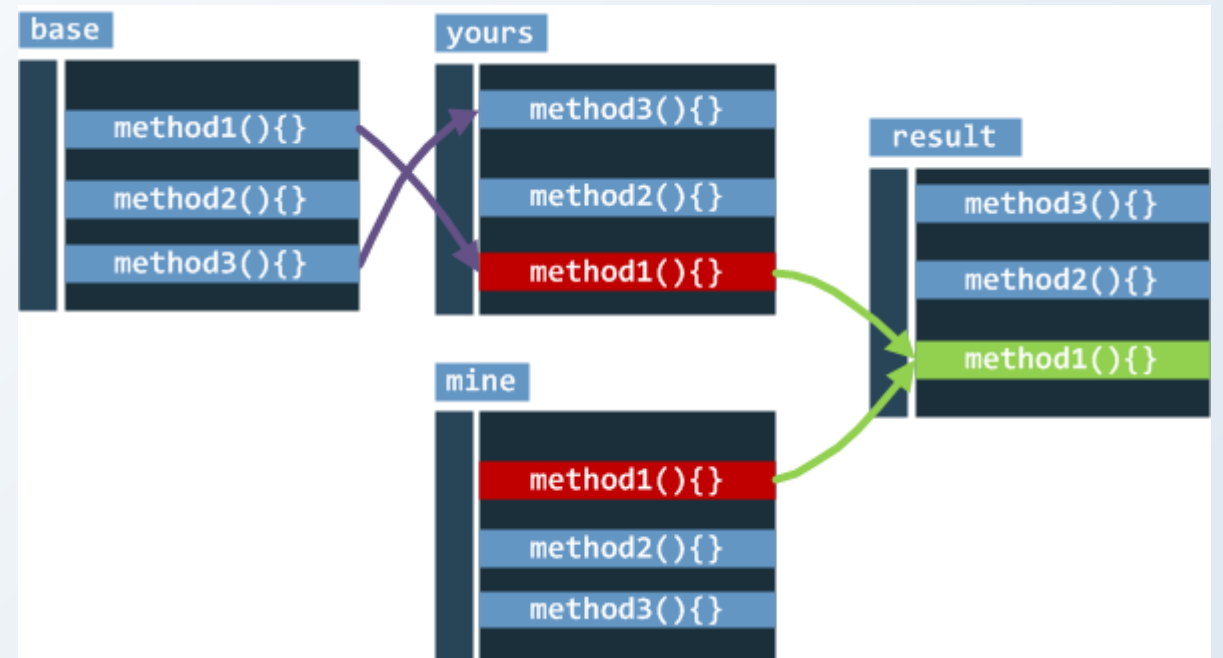


El merge a resolver



Fases del merge:

- Detección de conflictos a nivel de estructura de directorios.
- Detección de conflictos de ficheros.
- Resolución de conflictos dentro de ficheros.



Así surgen nuestras herramientas de merge:

1. Xmerge: basada en texto que sabe identificar fragmentos de código movido.
2. SemanticMerge: basada en estructura. El siguiente nivel!



SemanticMerge – cómo funciona

Es una herramienta dependiente del lenguaje a procesar

Entrada

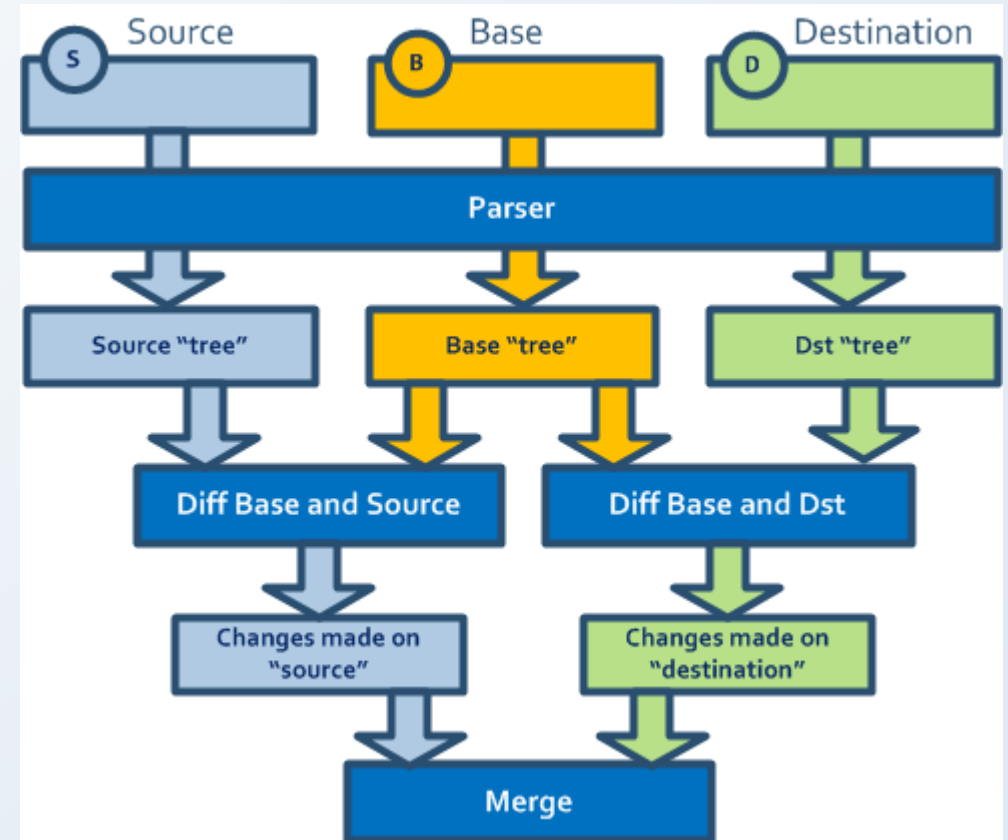
Las tres versiones del fichero, una por cada contribuidor.

Proceso

1. Parsear código fuente para obtener las estructuras.
2. Calcular diferencias entre las estructuras de código para obtener los cambios.
3. Realizar el merge de los cambios para obtener los conflictos si los hubiese.

Salida

Fichero resultado del merge.



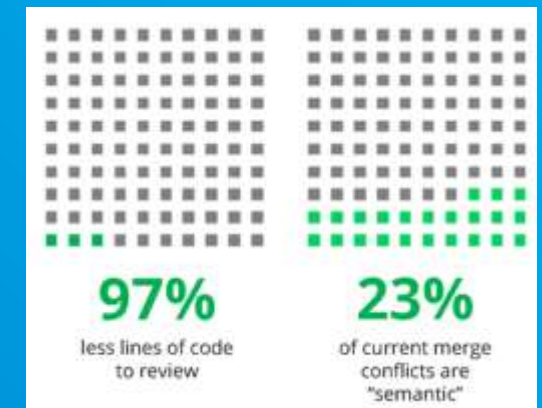
¿Por qué desarrollar un parser de C++?

La importancia de entender el código

- Motivo principal: entender el código para poder mezclarlo.
- ¿Qué nos aporta?
 - Mejorar las operaciones de diff y merge.
 - Facilita la realización de tareas que se evitan por miedo a los conflictos.
 - Incremento en la resolución de conflictos de forma automática.
 - Minimizar los tiempos de integración en un proyecto.

¿Se os ocurre alguna más?

¡Quizás sí! Lo veremos más adelante 😊





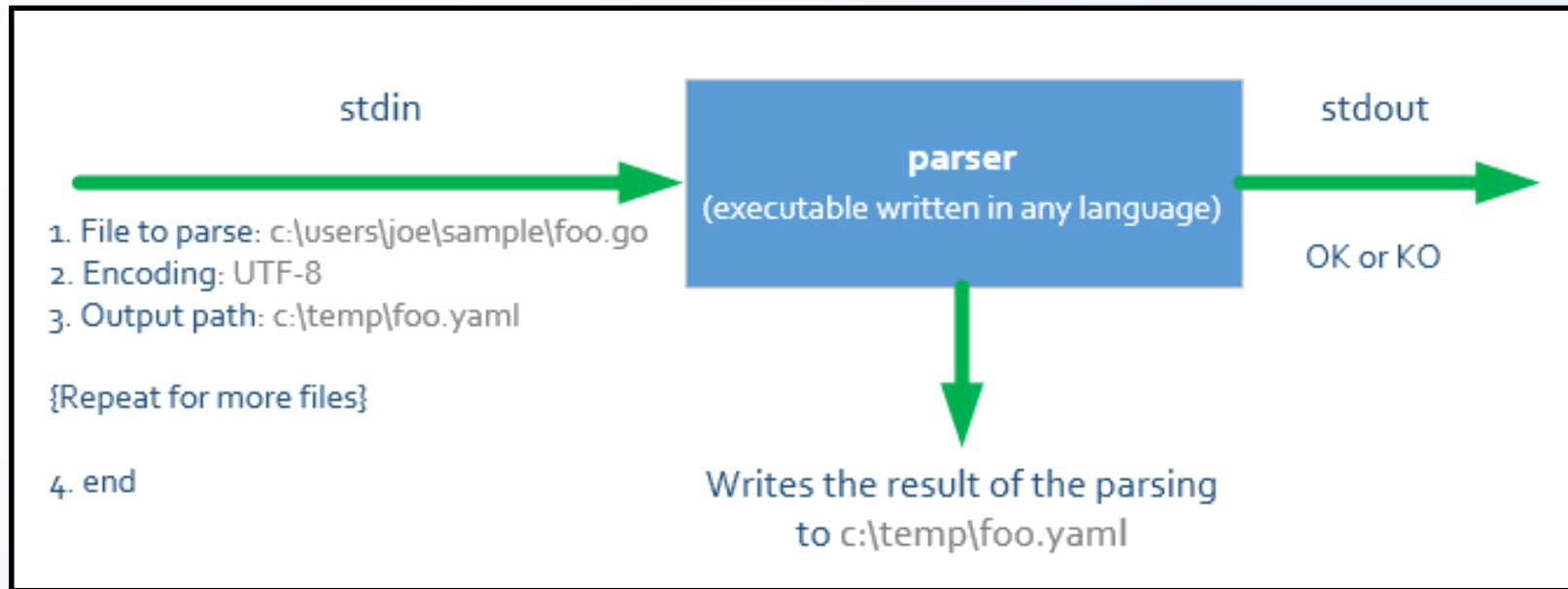
Parser de C++ basado en Eclipse CDT



@plasticscm @gmasterscm @semanticmerge

SemanticMerge - especificación de parsers

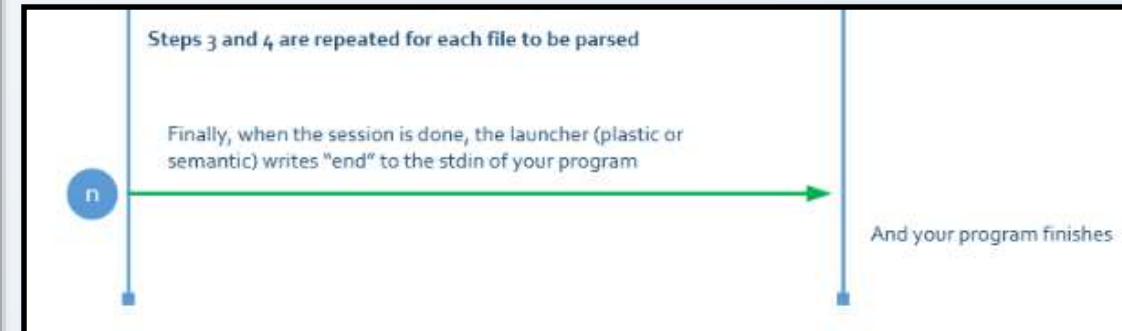
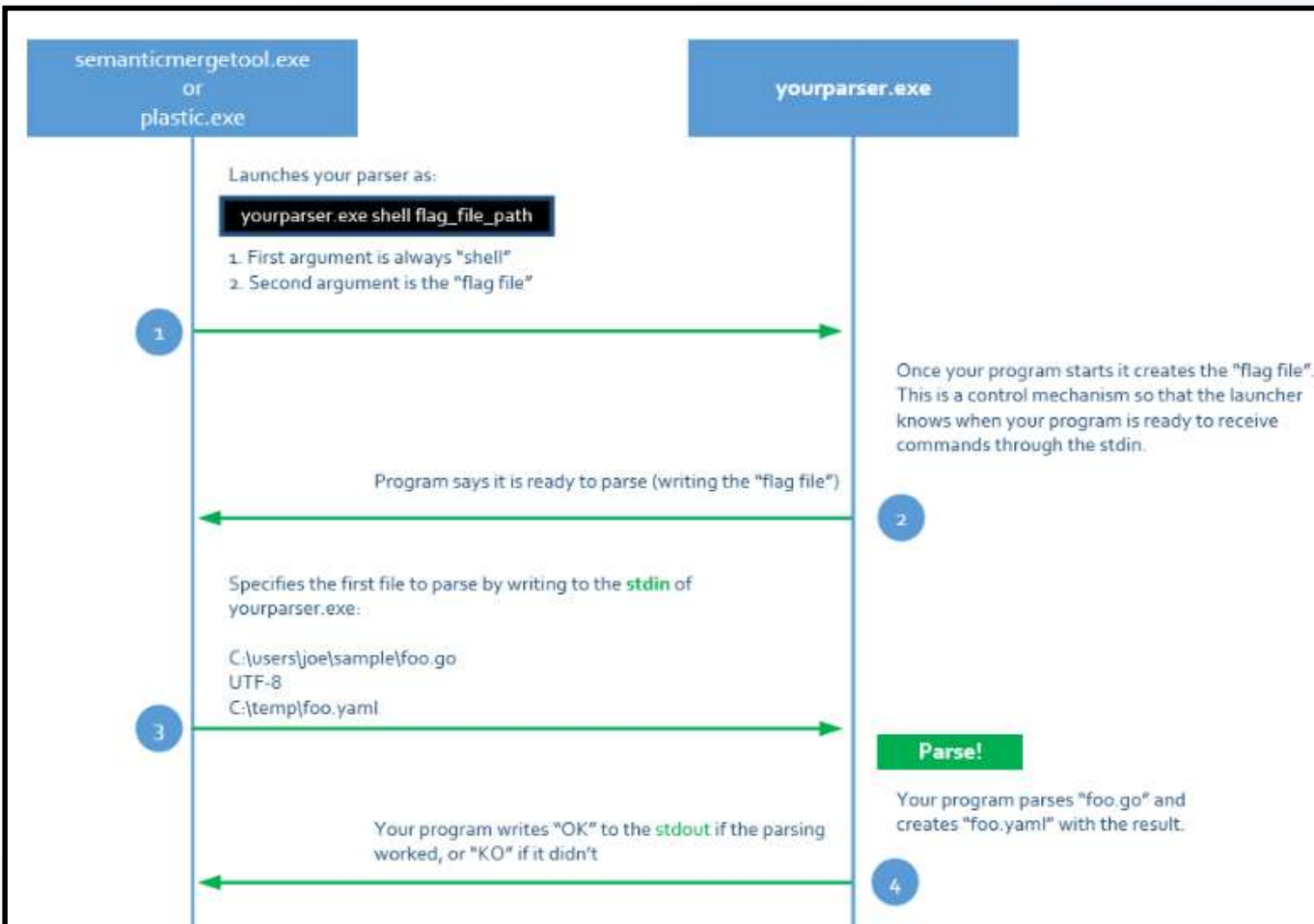
API para parser externos



Un parser externo es una aplicación de línea de comandos que se comunica con SemanticMerge a través de un API

SemanticMerge - especificación de parsers

¿Cómo es una sesión de parseo?

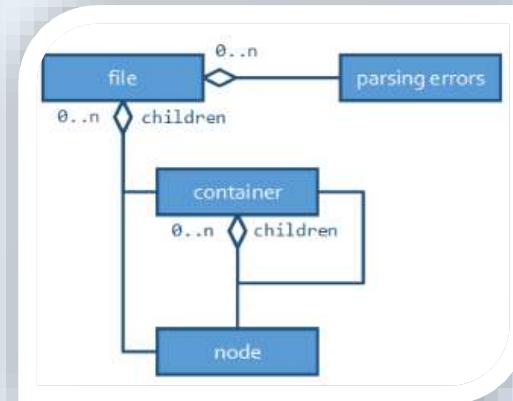


SemanticMerge - especificación de parsers

Cómo construir el árbol de declaraciones

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
span: [0, 17]	1	i	n	c	l	u	d	e	#	s	o	c	k	e	t	s	\r	\n						
span: [18, 34]	2	i	n	c	l	u	d	e	#	s	y	s	t	e	m	\r	\n							
Socket class	3	\r	\n																					
headerSpan : [35, 94]	4	/	/		t	h	i	s		c	o	m	m	e	n	t		i	s		f	o	r	
footerSpan : [354, 430]	5	c	l	a	s	s		S	o	c	k	e	t	\r	\n									
	6	{	\r	\n																				
	7	/	*		T	h	i	s		i	s		a		c	o	m	m	e	n	t			
	8	*		t	h	a	t		s	p	a	n	s		f	o	r		a					
	9	*	/	\r	\n																			
Connect()	10	v	o	i	d		C	o	n	n	e	c	t	(s	t	r	i	n	g				
span : [95, 275]	11	{	\r	\n																				
	12				S	o	c	k	e	t		L	i	b	r	a	r	y	.	C	o	n		
	13	}	\r	\n																				
	14	\r	\n																					
	15	v	o	i	d		D	i	s	c	o	n	n	e	c	t	()	\r	\n				
Disconnect()	16	{	\r	\n																				
span : [276, 353]	17				S	o	c	k	e	t		L	i	b	r	a	r	y	.	D	i	s		
	18	}	\r	\n																				
	19	\r	\n																					
	20	/	/		t	h	i	s		c	o	m	m	e	n	t								
	21	/	/		o	f		t	h	e		f	i	e	,									
	22	}	\r	\n																				

```
---
type: file
name: sample\source.code
locationSpan: {start: [1,0], end: [22,1]}
footerSpan: [0,-1]
parsingErrorsDetected: false
children:
  - type: include
    name: sockets
    locationSpan: {start: [1, 0], end: [1, 18]}
    span: [0, 17]
  - type: include
    name: system
    locationSpan: {start: [2, 0], end: [2,17]}
    span: [18, 34]
  - type: class
    name: Socket
    locationSpan: {start: [3,0], end: [22,1]}
    headerSpan: [35, 94]
    footerSpan: [354, 430]
    children:
      - type: method
        name: Connect
        locationSpan: {start: [7, 0], end: [13,6]}
        span: [95, 275]
      - type: method
        name: Disconnect
        locationSpan: {start: [14,0], end: [18,6]}
        span: [276, 353]
```



¿Por qué usar Eclipse CDT?

Evaluación de las distintas alternativas

- Parsers disponibles: C#, VB.net (Roslyn), Java (Eclipse JDT) y C (libclang).
- Pasaron unos años, hasta que en 2017... abordamos C++!
- Estudiamos distintas alternativas:
 - Antlr for C++, semanticdesigns, gccxml:
 - Fueron descartadas rápidamente (la gramática no era completa, necesidad de compilar para obtener el AST, etc.)
 - libclang:
 - **Problema:** el árbol AST que generaba no era completo porque no incluía los nodos con tipos desconocidos.
 - Tenía **dependencia** de los ficheros cabecera y de las librerías del sistema.
 - Nos encontramos con el clásico problema de parseo: `A B(C)` ¿declaración de función o definición de objeto?
 - Y es que C++ es un **lenguaje dependiente del contexto**.
 - El proceso de parseo y análisis semántico no se puede separar, cuando encuentra un tipo desconocido el nodo no se incluyen en el AST.
 - La **solución** pasaría por incluir las cabeceras para poder resolver los tipos.
 - En nuestro caso **no es viable** porque tendríamos que indicar dónde localizarlas y además estar disponibles.
 - En nuestra situación no es posible porque necesitamos los 3 contribuidores y no están en disco (control de versiones).



¿Por qué usar Eclipse CDT?

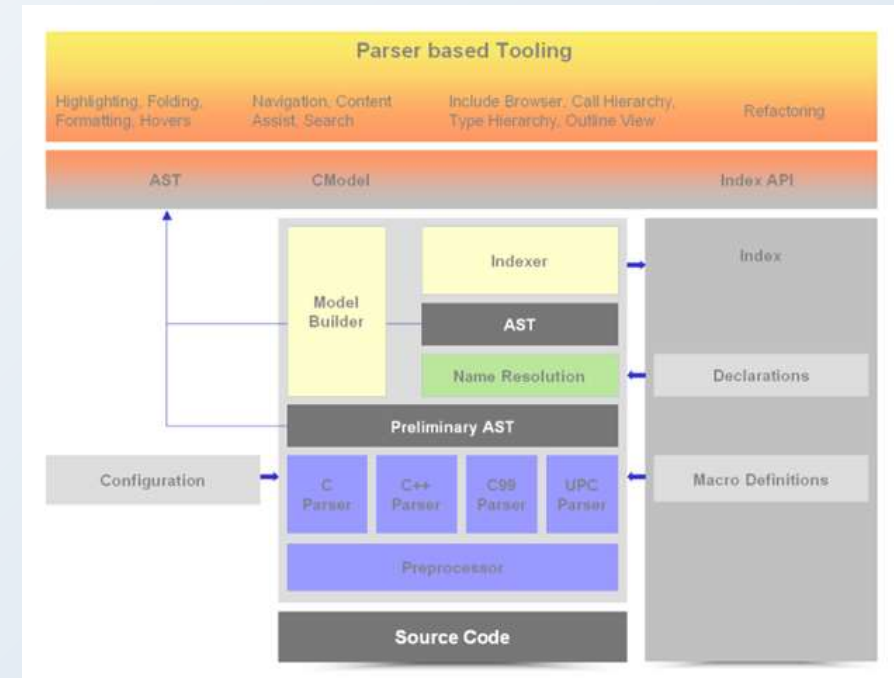
Evaluación de las distintas alternativas

- Eclipse C/C++ Development Toolkit (CDT):
 - Necesitamos el **análisis semántico sintáctico** del código para parsear todo el fichero y obtener su estructura.
 - Justo lo que hace Eclipse CDT, no es un compilador sino un IDE cuyo requisito principal es el **rendimiento**.
 - Es lo que permite al editor hacer folding, formatear un documento o mostrar la vista de outline.

Las fases incluyen:

1. Escaneo y preprocesamiento: convierte caracteres a tokens y se hace el preprocesamiento (expansión de macros, compilación condicional, inclusión de cabeceras)
2. Parseo: convierte los tokens en un AST + location map.

- https://wiki.eclipse.org/CDT/designs/Overview_of_Parsing
- https://wiki.eclipse.org/images/c/c7/CDT_APIs_for_code_introspection.pdf



Parser de C++ basado en Eclipse CDT

Cómo funciona

Conceptos

TranslationUnit: fichero de código fuente + cabeceras.

El árbol AST se puede analizar recorriéndolo de forma transversal (ASTVisitor).

Entrada

Fichero de código fuente + encoding.

Proceso

1. Crear el parser.
2. Obtener el TranslationUnit.
3. Crear el ASTVisitor y establecer propiedades.
4. Lanzar el recorrido a través del árbol:
 - Obtener el mapa de comentarios.
 - Obtener las macros y directivas de preprocesador.
 - Generar las declaraciones.
 - Obtener los errores de parseo.
5. Convertir las directivas condicionales en contenedores.

Salida

Fichero de resultado con el árbol de declaraciones.

```
PlasticVisitor
  PlasticVisitor(String, String)
  visit(IASTTranslationUnit) : int
  leave(IASTTranslationUnit) : int
  visit(ICPPASTNamespaceDefinition) : int
  leave(ICPPASTNamespaceDefinition) : int
  visit(IASTDeclaration) : int
  leave(IASTDeclaration) : int
```

```
public boolean process() {
    try {
        1 Parser parser = new Parser(mInputFilePath, mInputFileEncoding);
        if(parser.getCode() == null)
            return false;

        2 final IASTTranslationUnit translationUnit = parser.parse();

        3 PlasticVisitor visitor = new PlasticVisitor(mInputFilePath, parser.getCode());
        visitor.shouldVisitDeclarations = true;
        4 visitor.includeInactiveNodes = true;
        translationUnit.accept(visitor);

        5 FileDescriptor fileDescriptor = visitor.getFileDescriptor();
        ConditionalDirectiveConverter.convertToContainerDeclaration(
            fileDescriptor.getChildren());

        TreeDescriptorWriter descriptorWriter = new TreeDescriptorWriter(mOutputFilePath);
        descriptorWriter.write(fileDescriptor);
    } catch (Exception e) {
        return false;
    }
    return true;
}
```

```
mNodeCommentMap = ASTCommenter.getCommentedNodeMap(mTranslationUnit);
```

```
mTranslationUnit.getMacroDefinitions();
```

```
IASTProblem[] problems = mTranslationUnit.getPreprocessorProblems();
```



Parser de C++ basado en Eclipse CDT

Problemas encontrados

Macros

- Hay que manejar los distintos escenarios:
¿están dentro de una declaración? ¿definen una declaración?

```
static void __init func(void) /* macro inside the function header */  
{  
}  
  
MACHINE_START(W90P910EVB, "W90P910EVB")  
/* struct declaration */  
/* Maintainer: Wan ZongShun */  
.phys_io      = W90X900_PA_UART,  
.io_pg_offst   = (((u32)W90X900_VA_UART) >> 18) & 0xfffc,  
.boot_params   = 0,  
MACHINE_END
```

Directivas condicionales

- Son un grupo contenedor.
- Hay que procesar los nodos inactivos.

```
#if PY_MAJOR_VERSION < 3  
void func(int, int, int) {  
    process(a);  
}  
  
void func2(int, int)  
{  
    process(a);  
}  
#else  
void func(int) {  
#endif  
    unprocess(b);  
}
```

Comentarios

- Los comentarios son parte de la declaración.

Detección de errores de parseo

- Por ejemplo, si falta un ';' al final de una instrucción



Parser de C++ basado en Eclipse CDT

Bonus track!

Regla de oro

Cualquier carácter del fichero tiene que estar contenido en una declaración de forma que a partir del conjunto de las mismas se pueda construir el fichero resultado.

Check salva-vidas :P

- Comprobación de reconstrucción del fichero a partir del árbol de declaraciones, en caso de detectar algún problema se lanza Xmerge.

Testing

- Millones de ficheros parseados de los repositorios más relevantes de GitHub.

Siguientes pasos

- Formateador de declaraciones para automatizar conflictos por cambios de formato.





Aplicaciones prácticas del parser

- Herramienta de merge: **semanticmerge**
- Control de versiones: **plasticscm** y **gmaster**
- Bot que resuelve conflictos de PR sin intervención manual: **mergedroid**



SemanticMerge (srcdiffs)

semanticdiff from base.cpp and src.cpp

Current: 1/3

Semantic Outline

- Moved - 1 item(s)
 - [function] Date::print moved to Date and renamed to print
- Changed - 1 item(s)
 - [function] main
- Renamed - 1 item(s)
 - [function] Date::print moved to Date and renamed to print

C:\Users\violeta\AppData\Local\semanticmerge\samples\CPlusPlus\1. Date\base.cpp

```
10
11 public:
12   Date(int, int, int);
13   void set(int, int, int);
14   void print();
15
16 };
17
18 int main()
19 {
20   Date today(1, 1, 2017);
21
22   cout << "This program was written on ";
23   today.print();
24
25   cout << "This program was modified on ";
26   today.set(2, 1, 2017);
27   today.print();
28
29   return 0;
30 }
31
32 Date::Date(int d, int m, int y)
33 {
34   if(d > 0 && d < 31) day = d;
35   if(m > 0 && m < 13) month = m;
36   if(y > 0) year = y;
37 }
38
39 void Date::print()
40 {
41   cout << day << "-" << month << "-" << year << endl;
42 }
43
44 void Date::set(int d, int m, int y)
45 {
46   if(d > 0 && d < 31) day = d;
47   if(m > 0 && m < 13) month = m;
48   if(y > 0) year = y;
49 }
50
51
```

C:\Users\violeta\AppData\Local\semanticmerge\samples\CPlusPlus\1. Date\src.cpp

```
9   int day;
10
11 public:
12   Date(int, int, int);
13   void set(int, int, int);
14   void print();
15
16 void print()
17 {
18   cout << day << "-" << month << "-" << year << endl;
19 }
20
21 };
22
23 int main()
24 {
25   Date today(1, 1, 2017);
26
27   cout << "This program was written on ";
28   today.print();
29
30   cout << "This program was modified on ";
31   today.set(18, 1, 2017);
32   today.print();
33
34   return 0;
35 }
36
37 Date::Date(int d, int m, int y)
38 {
39   if(d > 0 && d < 31) day = d;
40   if(m > 0 && m < 13) month = m;
41   if(y > 0) year = y;
42 }
43
44 void Date::set(int d, int m, int y)
45 {
46   if(d > 0 && d < 31) day = d;
47   if(m > 0 && m < 13) month = m;
48   if(y > 0) year = y;
49 }
50
```

Visual diff Run text diff Restart diff



SemanticMerge (dstdiffs)

semanticdiff from base.cpp and dst.cpp

Current: 1/3

Semantic Outline

- Moved - 1 item(s)
 - [function] main moved down 2 positions
- Changed - 2 item(s)
 - [function] Date::print
 - [function] main

C:\Users\violeta\AppData\Local\semanticmerge\samples\CPlusPlus\1. Date\base.cpp

```
9  ..int day;
10
11 public:
12  ..Date(int, int, int);
13  ..void set(int, int, int);
14  ..void print();
15
16 };
17
18 [C M] int main()
19 {
20  ...Date today(1, 1, 2017);
21
22  ...cout << "This program was written on ";
23  ...today.print();
24
25  ...cout << "This program was modified on ";
26  ...today.set(2, 1, 2017);
27  ...today.print();
28
29  ...return 0;
30 }
31
32 Date::Date(int d, int m, int y)
33 {
34  ..if(d>0 && d<31) day = d;
35  ..if(m>0 && m<13) month = m;
36  ..if(y>0) year = y;
37 }
38
39 [C] void Date::print()
40 {
41  ..cout << day << "/" << month << "/" << year << endl;
42 }
43
44 void Date::set(int d, int m, int y)
45 {
46  ..if(d>0 && d<31) day = d;
47  ..if(m>0 && m<13) month = m;
48  ..if(y>0) year = y;
49 }
50
```

C:\Users\violeta\AppData\Local\semanticmerge\samples\CPlusPlus\1. Date\dst.cpp

```
9  ..int day;
10
11 public:
12  ..Date(int, int, int);
13  ..void set(int, int, int);
14  ..void print();
15
16 };
17
18 Date::Date(int d, int m, int y)
19 {
20  ..if(d>0 && d<31) day = d;
21  ..if(m>0 && m<13) month = m;
22  ..if(y>0) year = y;
23 }
24
25 [C] void Date::print()
26 {
27  ..cout << day << "/" << month << "/" << year << endl;
28 }
29
30 [C M] int main()
31 {
32  ...Date today(1, 1, 2017);
33
34  ...cout << "This program was written on ";
35  ...today.print();
36
37  ...cout << "This program was modified on ";
38  ...today.set(31, 1, 2017);
39  ...today.print();
40
41  ...return 0;
42 }
43
44 void Date::set(int d, int m, int y)
45 {
46  ..if(d>0 && d<31) day = d;
47  ..if(m>0 && m<13) month = m;
48  ..if(y>0) year = y;
49 }
50
```

Visual diff Run text diff Restart diff



SemanticMerge (merge!)

SemanticMerge (merge!)

Save & Exit | 1/1 - Conflicts to solve: 1

Semantic Outline

- Conflicts to solve (1)
 - [function] main
- main changed on both contributors
 - changed on source
 - changed on destination
 - merge
- Src - their changes (2)
 - Moved - 1 item(s)
 - [function] Date::print moved to Date and renamed to print
 - Changed - 1 item(s)
 - [function] main
 - Renamed - 1 item(s)
 - [function] Date::print moved to Date and renamed to print
 - visual diff run text diff
- Dst - your changes (3)
 - Moved - 1 item(s)
 - [function] main moved down 2 positions
 - Changed - 2 item(s)
 - [function] Date::print
 - [function] main
 - visual diff run text diff

Source: C:\Users\violeta\AppData\Local\semanticmerge\samples\CPlusPlus\1.Date\src.c

```
18 --cout<<"day<<"-"<<"month<<"-"<<"year<<"endl;
19 }
20
21 };
22
23 int main()
24 {
25     Date today(1,1,2017);
26
27     cout<<"This program was written on ";
28     today.print();
29
30     cout<<"This program was modified on ";
31     today.set(18,1,2017);
32     today.print();
33
34     return 0;
35 }
```

Base: C:\Users\violeta\AppData\Local\semanticmerge\samples\CPlusPlus\1.Date\base.c

```
13 --void set(int, int, int);
14 --void print();
15
16 };
17
18 int main()
19 {
20     Date today(1,1,2017);
21
22     cout<<"This program was written on ";
23     today.print();
24
25     cout<<"This program was modified on ";
26     today.set(2,1,2017);
27     today.print();
28
29     return 0;
30 }
```

Destination: C:\Users\violeta\AppData\Local\semanticmerge\samples\CPlusPlus\1.Date\

```
25 void Date::print()
26 {
27     cout<<"day<<"-"<<"month<<"-"<<"year<<"endl;
28 }
29
30 int main()
31 {
32     Date today(1,1,2017);
33
34     cout<<"This program was written on ";
35     today.print();
36
37     cout<<"This program was modified on ";
38     today.set(31,1,2017);
39     today.print();
40
41     return 0;
42 }
43
44 void Date::set(int d, int m, int y)
45 {
46     if(d>0 && d<31, d==d;
47     if(m>0 && m<13, m=m;
48     if(y>0, year=y;
49 }
```

Visual Merge Run text merge Restart merge Process all merges



PlasticSCM

The screenshot displays the Plastic SCM application interface. The top bar shows the workspace name "mainrep-TestStatusOutput-Client" and the branch "/main/cplusplus". The "Workspace Explorer" tab is active, showing "Pending changes (1 Item - 1 checked)". Below this, a table lists the changes, with "Date.cpp" marked as "Checked-out". The "Semantic Outline" pane on the left shows a tree view of the project structure, including "Moved - 1 item(s)", "Changed - 2 item(s)", and "Renamed - 1 item(s)". The main editor area displays the code for "Date.cpp", with a diff view showing changes between the current workspace and the repository. The diff view highlights changes in the "main" branch, showing a "void print()" function and a "main" function. The bottom status bar indicates the current revision is "1/4".

Plastic SCM - violeta - Pending changes (1 Item - 1 checked)

mainrep-TestStatusOutput-Client Branch /main/cplusplus mainrep-TestStatusOutput-Client localhost:8084 Workspace path c:\mainrep-TestStatusOutput-Client

Workspace Explorer Pending changes (1 Item - 1 checked)

Checkin Undo changes Options Hide giffs Check/Clear all Filter:

Item Status

Changed items - 1 of 1 items selected

Date.cpp Checked-out

Current: 1/4 Options Find...

Semantic Outline /Date.cpp#cs:4@br:/main/cplusplus violeta

```
11 public:
12   Date(int, int, int);
13   void set(int, int, int);
14   void print();
15
16 };
17
18 int main()
19 {
20   Date today(1, 1, 2017);
21
22   cout << "This program was written on ";
23   today.print();
24
25   cout << "This program was modified on ";
26   today.set(2, 1, 2017);
27   today.print();
28
29   return 0;
30 }
31
32 Date::Date(int d, int m, int y)
33 {
```

/Date.cpp#cs:4@br:/main/cplusplus violeta Workspace Revision

```
16 void print()
17 {
18   cout << day << "-" << month << "-" << year << endl;
19 }
20
21 };
22
23 int main()
24 {
25   Date today(1, 1, 2017);
26
27   cout << "This program was written on ";
28   today.print();
29
30   cout << "This program was modified on ";
31   today.set(18, 1, 2017);
32   today.print();
33
34   return 0;
35 }
36
37 Date::Date(int d, int m, int y)
38 {
```

Text diff Semantic diff Visual diff Skip format changes Reformat



gmaster

The screenshot displays the gmaster IDE interface. The top bar shows the repository name 'multifileRepo - master - gmaster (Beta)' and commit statistics '[GC: 74MB Committed: 300MB]'. The left sidebar contains navigation options: Repositories, multifileRepo, master, Push, Pull, Fetch, Commit (2), Branch Explorer, Commit History, and Remotes. The main workspace is divided into several panels. The top panel shows filters for 'only relevant', 'remotes', and 'last year (20/02/2018)'. Below this is a commit selection area for 'Commit e79aa56' with buttons for 'Create branch', 'Checkout', 'Tag', and 'Merge from'. The central panel displays a diff view of 'Date.cpp', comparing the current state with the parent commit. The diff shows changes in the 'main' function, including modifications to the 'today' variable and the 'set' method. The right sidebar contains 'Commit properties' (Sha: e79aa56, Date: 20/02/2019 19:31, Author: vsanchezm<vsanchezm@codiceso>) and 'Changed files' (Parent sha 6cfc964: 1 item(s), Modified: 1 item(s), Date.cpp). The bottom status bar indicates 'Differences' and 'File blame - Date.cpp (e79aa56)'. The bottom right corner has buttons for 'Send feedback' and 'Tour'.

multifileRepo - master - gmaster (Beta) [GC: 74MB Committed: 300MB]

Repositories multifileRepo master Push Pull Fetch

Commit 2

Branch Explorer

Commit History

Remotes

Filters: only relevant remotes last year (20/02/2018)

20/02/2019

master

Commit e79aa56: Create branch Checkout Tag Merge from Check the context menu for more options and filters

Differences

Current: 1/2 Options Find...

Semantic Outline

- Moved - 1 item(s)
 - [function] main moved down 1 position
- Changed - 1 item(s)
 - [function] main

Date.cpp

```
19 }
20
21 };
22
23 C M int main()
24 {
25     Date today(1,1,2017);
26
27     cout<<"This program was written on ";
28     today.print();
29
30     cout<<"This program was modified on ";
31     today.set(18,1,2017);
32     today.print();
33
34     return 0;
35 }
36
37 Date::Date(int d, int m, int y)
38 {
39     if(d>0 && d<31) day = d;
```

Date.cpp

```
23 Date::Date(int d, int m, int y)
24 {
25     if(d>0 && d<31) day = d;
26     if(m>0 && m<13) month = m;
27     if(y>0) year = y;
28 }
29
30 C M int main()
31 {
32     Date today(1,1,2017);
33
34     cout<<"This program was written on ";
35     today.print();
36
37     cout<<"This program was modified on ";
38     today.set(19,1,2017);
39     today.print();
40
41     return 0;
42 }
43
```

Text diff Semantic diff Visual diff Skip format changes Reformat

Differences File blame - Date.cpp (e79aa56)

Commit properties

Sha: e79aa56 | Date: 20/02/2019 19:31

Author: vsanchezm<vsanchezm@codiceso>

Change date file!

Head of branch master

Changed files

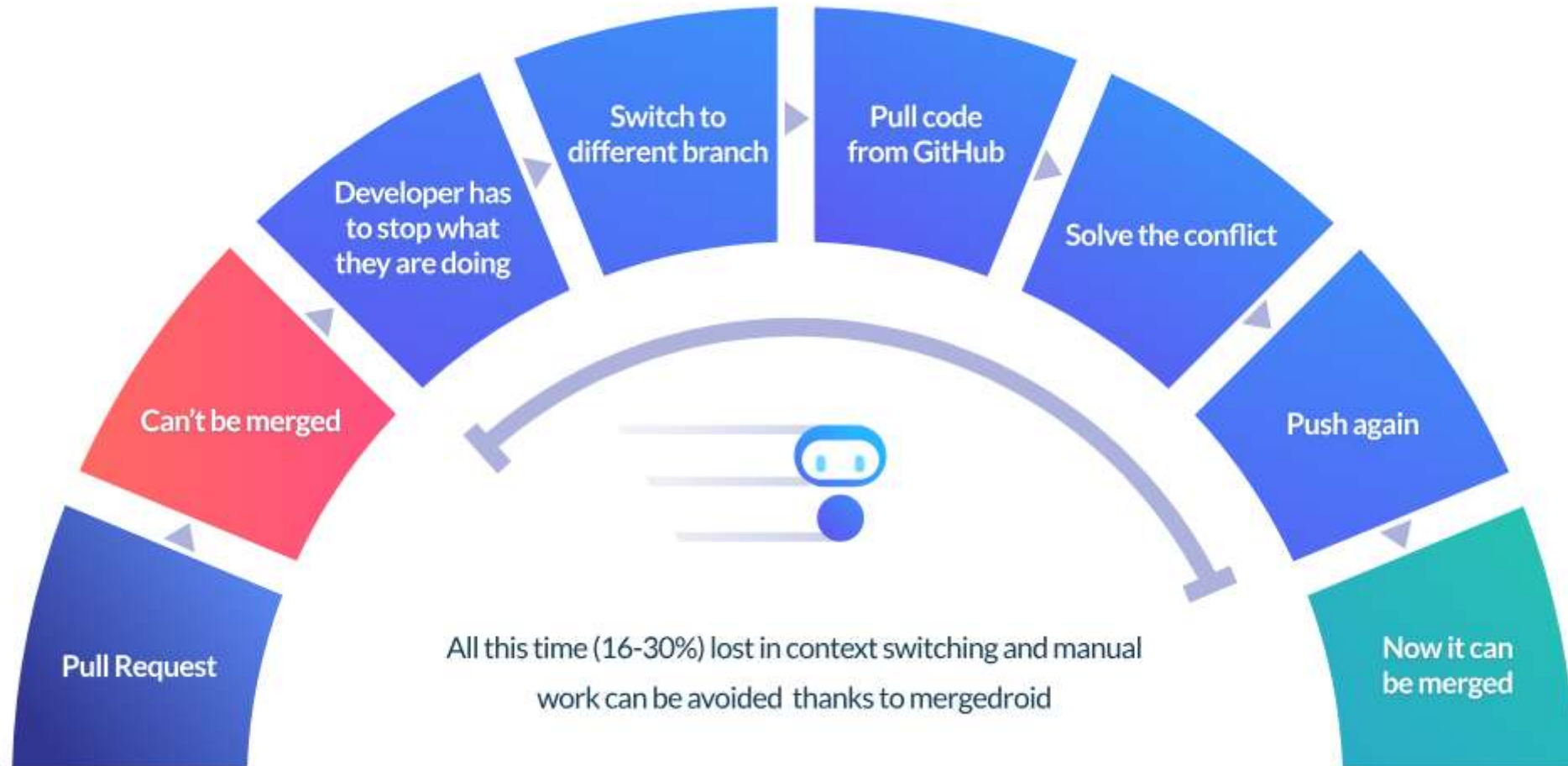
- Parent sha 6cfc964: 1 item(s) M 1
- Modified: 1 item(s)
 - Date.cpp 8 8 Date.cpp

Ready

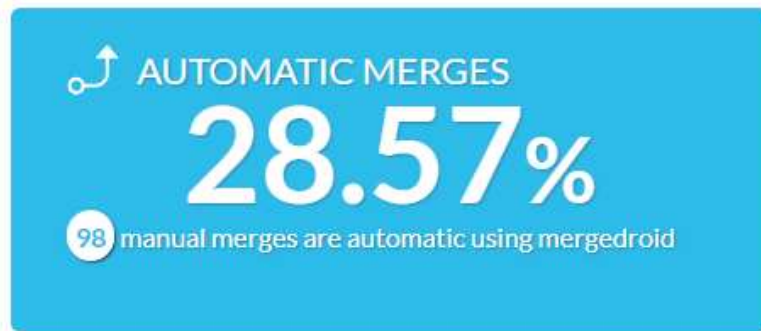
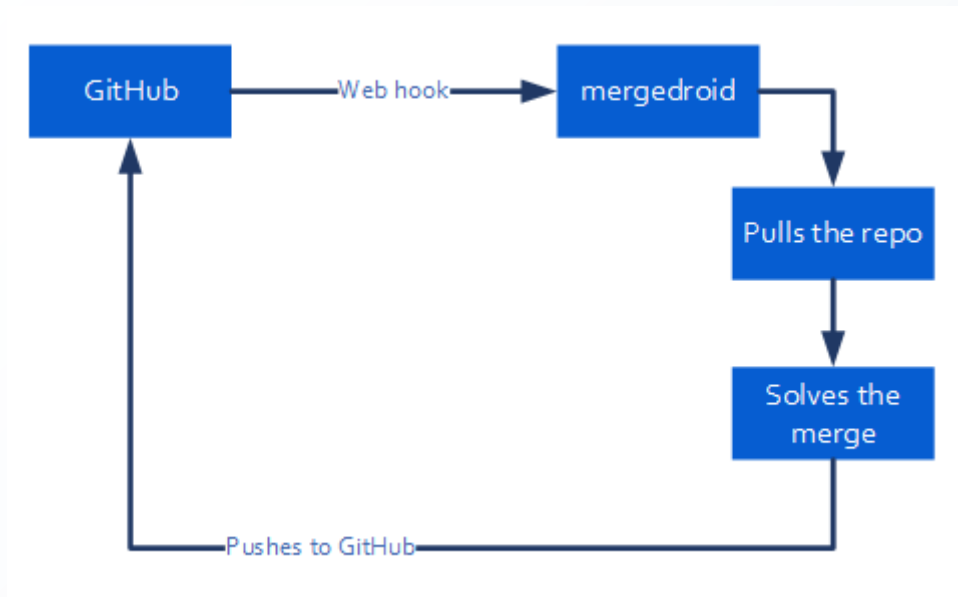
Send feedback Tour



mergedroid



mergedroid



<https://github.com/gmasterscm/gitextensions>

Conversation 0 Commits 2 Files changed 2

miryamGSM commented 6 days ago Owner

Pull request to integrate a multi-file refactor ;)

bruiz and others added some commits 24 days ago

- MV NotifyLicenseError from foo to bar 80e3468
- Automatic merge thanks to the power of MergeDroid 5d500f8

Add more commits by pushing to the **task001-other** branch on miryamGSM/m-testcases.

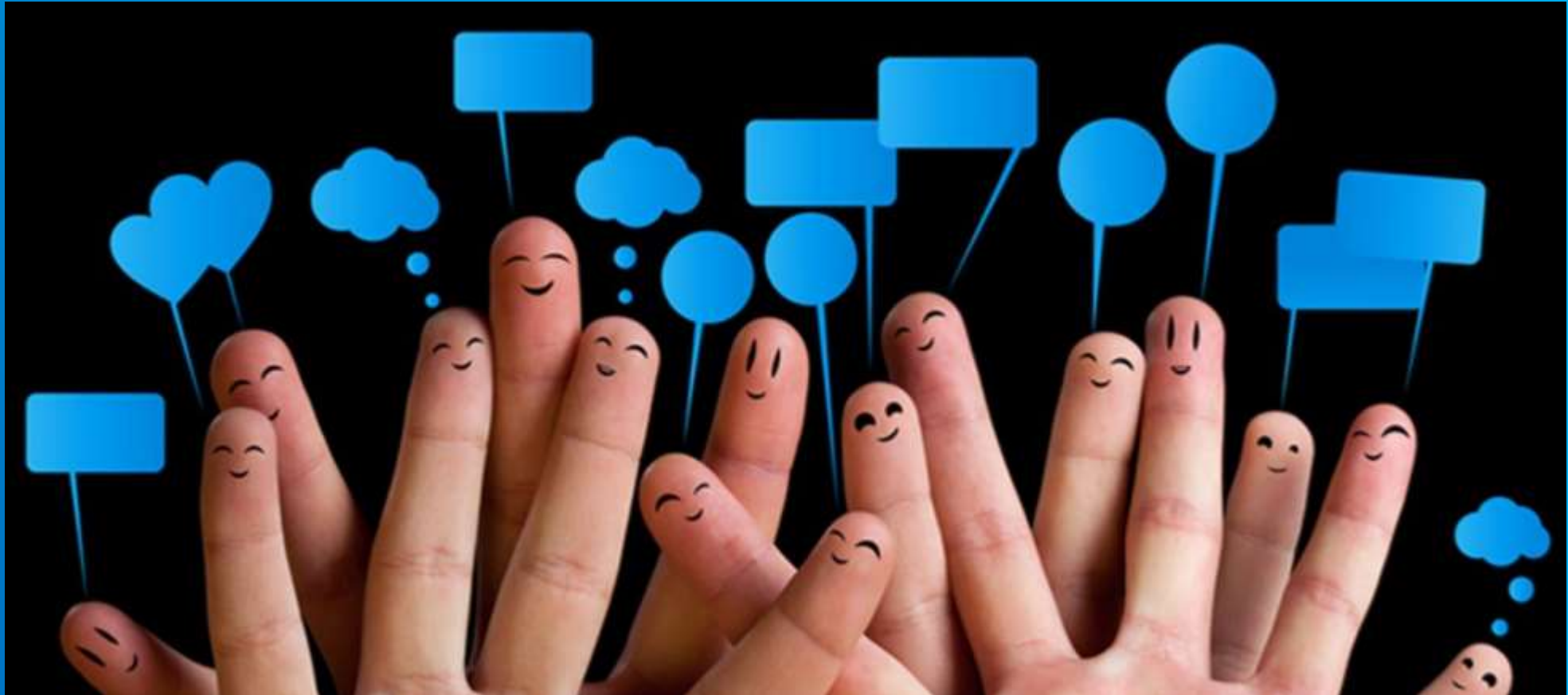
This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



@plasticscm @gmasterscm @semanticmerge

Comentarios / Otras experiencias



THANKS
FOR YOUR
TIME

