



Improving the Optimized Debugging Experience

Orlando Cazalet-Hyams, Sony Interactive Entertainment
LLVM Developers' Meeting
October 2019

Motivation

Debugging optimized code is terrible (C++)

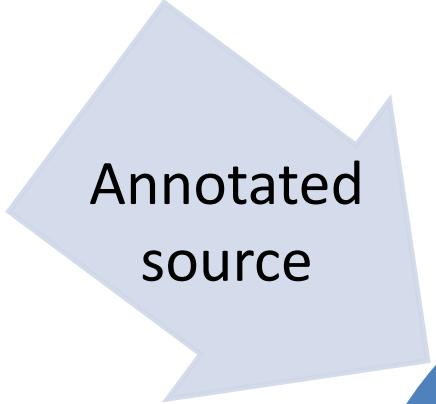
Debuggers lie to developers

Compiler is to blame

There's a lot we can do



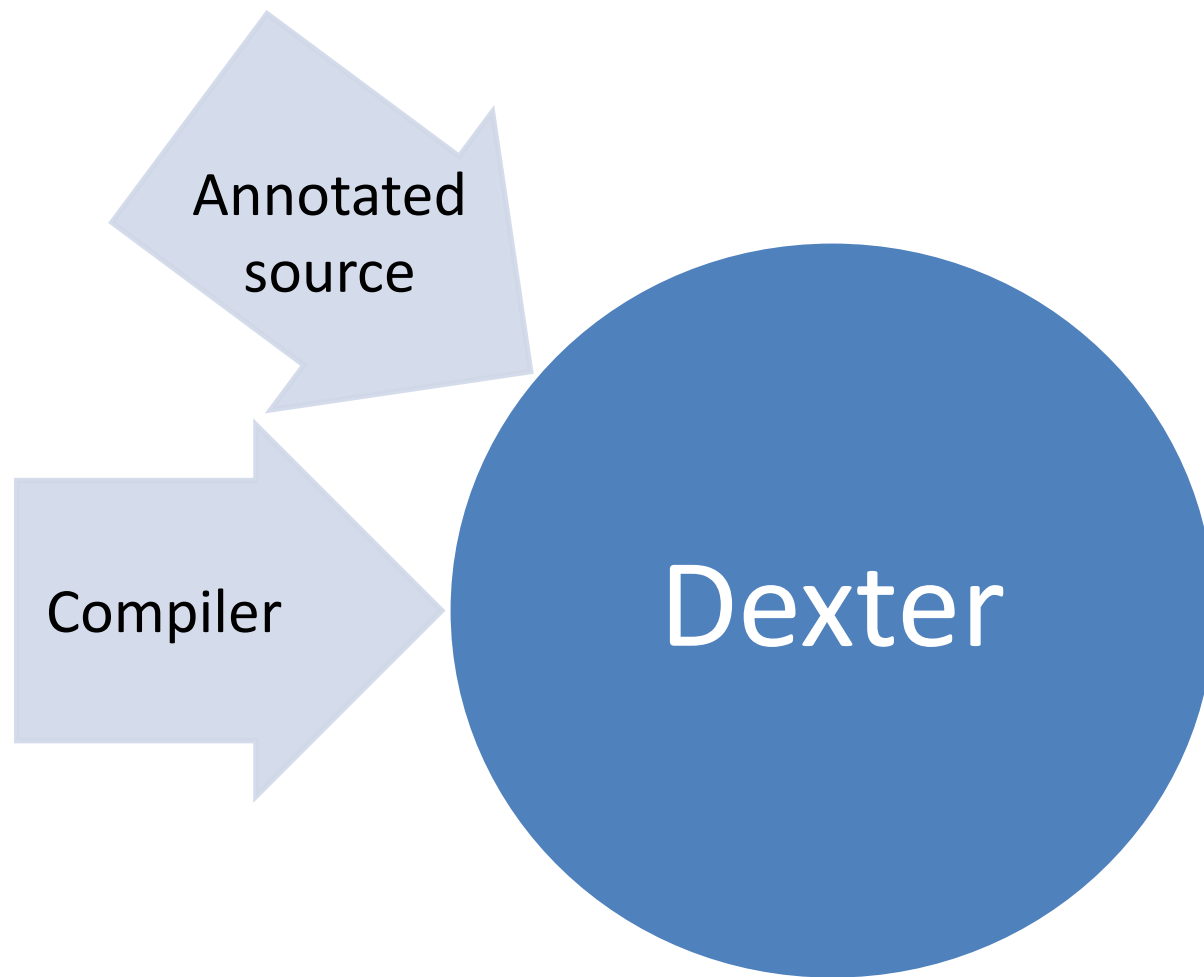
Dexter

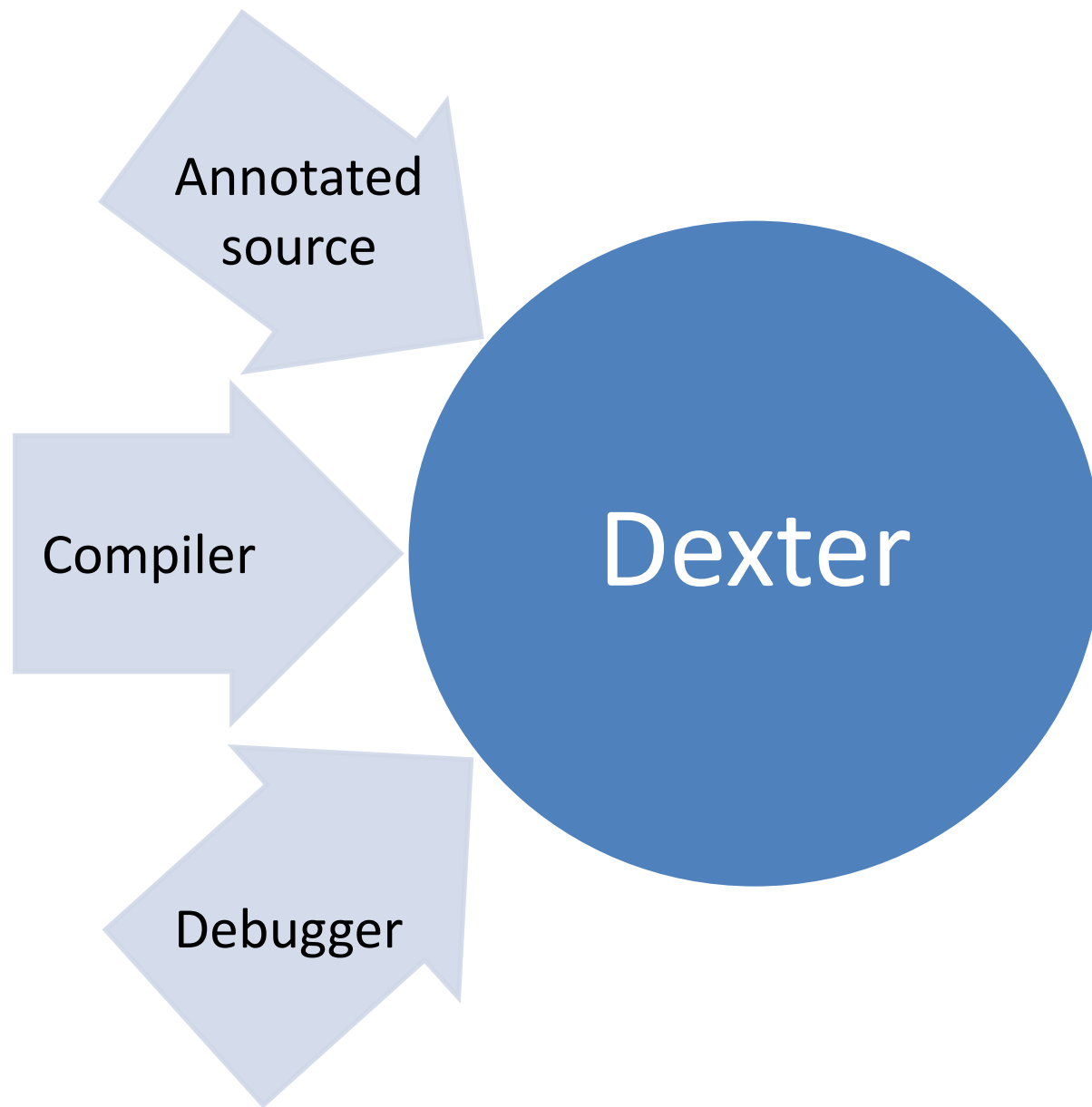


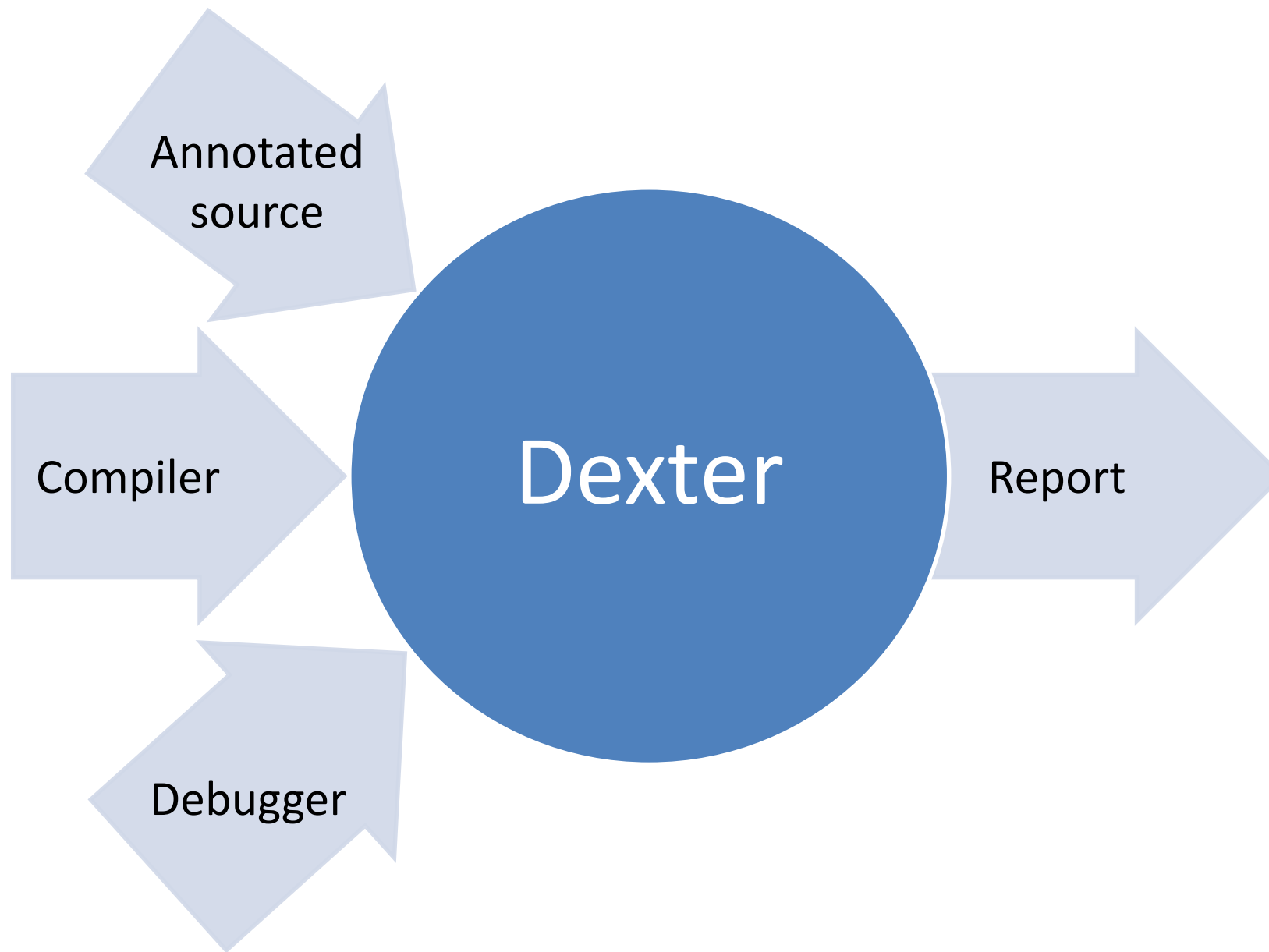
Annotated
source



Dexter







Workflow

Pick area to interrogate

Write tests

Run Dexter

Investigate results

Report and fix bugs

An example bug

Incorrect value for variable

Debugging expectations

<code>z = 5;</code>	<code>z == ?</code>
<code>x = 6;</code>	<code>z == 5</code>
<code>if (cond) {</code> <code> z = x;</code>	
<code> foo(z);</code>	<code>z == 6</code>
<code>}</code>	
<code>return;</code>	

Optimized build report

<code>z = 5;</code>	<code>z == ?</code>
<code>x = 6;</code>	<code>z == 5</code>
<code>if (cond) { z = x; foo(z); }</code>	<code>z == 6</code>
<code>return;</code>	

CodeGenPrepare::placeDbgValues

Bugzilla 38754

`dbg.value(<location>, <variable>, ...)`

dbg.value(<location>, <variable>, ...)

dbg.value(<location>, <variable>, ...)

Redundant assignment

```
z = 5;  
x = 6;  
  
if (cond) {  
    z = x;  
    foo(z);  
}  
return;
```


Optimized away

```
z = 5;  
x = 6;  
  
if (cond) {  
    z = x;  
}  
foo(z);  
return;
```

```
z = 5;  
x = 6;  
  
if (cond) {  
    dbg.value (&x, z);  
}  
foo(x);  
return;
```

CodeGenPrepare

```
z = 5;  
x = 6;  
  
if (cond) {  
    z = x;  
}  
foo(z);  
return;
```

```
z = 5;  
x = 6;  
  
if (cond) {  
    dbg.value (&x, z);  
}  
foo(x);  
return;
```

```
z = 5;  
x = 6;  
dbg.value (&x, z);  
  
if (cond) {  
    foo(x);  
}  
return;
```

The bugs

Umbrella Bugzilla 38768

More about Dexter

Explore	Test
Fast edit-compile-debug	Regression test suite
Debugger agnostic commands	

Links

Dexter RFC

<http://lists.lvm.org/pipermail/llvm-dev/2019-October/135773.html>

Umbrella bug

https://bugs.lvm.org/show_bug.cgi?id=38768

The end

Thank you