Open Multimethods for C++

Peter Pirkelbauer Yuriy Solodkyy Bjarne Stroustrup

Texas A&M University

Generative Programming and Component Engineering

Parasol
Smarter computing.
Texas A&M University

Overview

- Motiviation and problem description
- Design space
- Implementation details
- Results

Parasol
Smarter computing.
Texas A&M University

Single Dispatch (virtual function call)

Cornerstone of OOP languages

- Invoked foo depends on the dynamic type of the receiver (a)

```
struct A {
  virtual void foo() { cout << "Good afternoon"; }
};

struct B : A {
  virtual void foo() { cout << "Guten Tag"; }
};

void bar(A& obj) {
  obj.foo();
}
```

Parasol
Smarter computing.
Texas A&M University

Open Multi Dispatch

- Invoked foo depends on the dynamic type of both arguments (a,b)

```
struct A { virtual ~A(); };
struct B : A {};

void foo(virtual A& x, virtual A& y) { cout << "AA"; }
void foo(virtual B& x, virtual A& y) { cout << "BA"; }
void foo(virtual B& x, virtual B& y) { cout << "BB"; }

void bar(A& obj1, A& obj2) {
   foo(obj1, obj2);
}
```

- Dispatch Behaviour:

$1^{st} \backslash 2^{nd}$	A	B
A	AA	AA
B	BA	BB

Parasol
Smarter computing.
Texas A&M University

Open Multi Dispatch

- Invoked foo depends on the dynamic type of both arguments (a,b)

```
struct A { virtual ~A(); };
struct B : A {};

void foo(virtual A& x, virtual A& y) { cout << "AA"; }
void foo(virtual B& x, virtual A& y) { cout << "BA"; }
void foo(virtual B& x, virtual B& y) { cout << "BB"; }

void bar(A& obj1, A& obj2) {
   foo(obj1, obj2);
}
```

- Dispatch Behaviour:

$1^{st} \backslash 2^{nd}$	A	B
A	AA	AA
B	BA	BB

Parasol
Smarter computing.
Texas A&M University

Open Multi Dispatch

- Invoked foo depends on the dynamic type of both arguments (a,b)

```
struct A { virtual ~A(); };
struct B : A {};

void foo(virtual A& x, virtual A& y) { cout << "AA"; }
void foo(virtual B& x, virtual A& y) { cout << "BA"; }
void foo(virtual B& x, virtual B& y) { cout << "BB"; }

void bar(A& obj1, A& obj2) {
   foo(obj1, obj2);
}
```

- Dispatch Behaviour:

$1^{st} \backslash 2^{nd}$	A	B
A	AA	AA
B	BA	BB

Parasol
Smarter computing.
Texas A&M University

Open Multi Dispatch

- Invoked foo depends on the dynamic type of both arguments (a,b)

```
struct A { virtual ~A(); };
struct B : A {};

void foo(virtual A& x, virtual A& y) { cout << "AA"; }
void foo(virtual B& x, virtual A& y) { cout << "BA"; }
void foo(virtual B& x, virtual B& y) { cout << "BB"; }

void bar(A& obj1, A& obj2) {
   foo(obj1, obj2);
}
```

- Dispatch Behaviour:

$1^{st} \backslash 2^{nd}$	A	B
A	AA	AA
B	BA	BB

Parasol
Smarter computing.
Texas A&M University

Contributions

- Integration of runtime dispatch with overload resolution
- Open Class Extensions
  - Separates classes from virtual functions
- Multimethods semantics for the C++Object Model
- Use covariant return type for ambiguity resolution
- Efficient implementation
  - EDG based modification of the IA-64 Object Model
  - Prelinker
  - Overhead is ~12% of a regular virtual function call

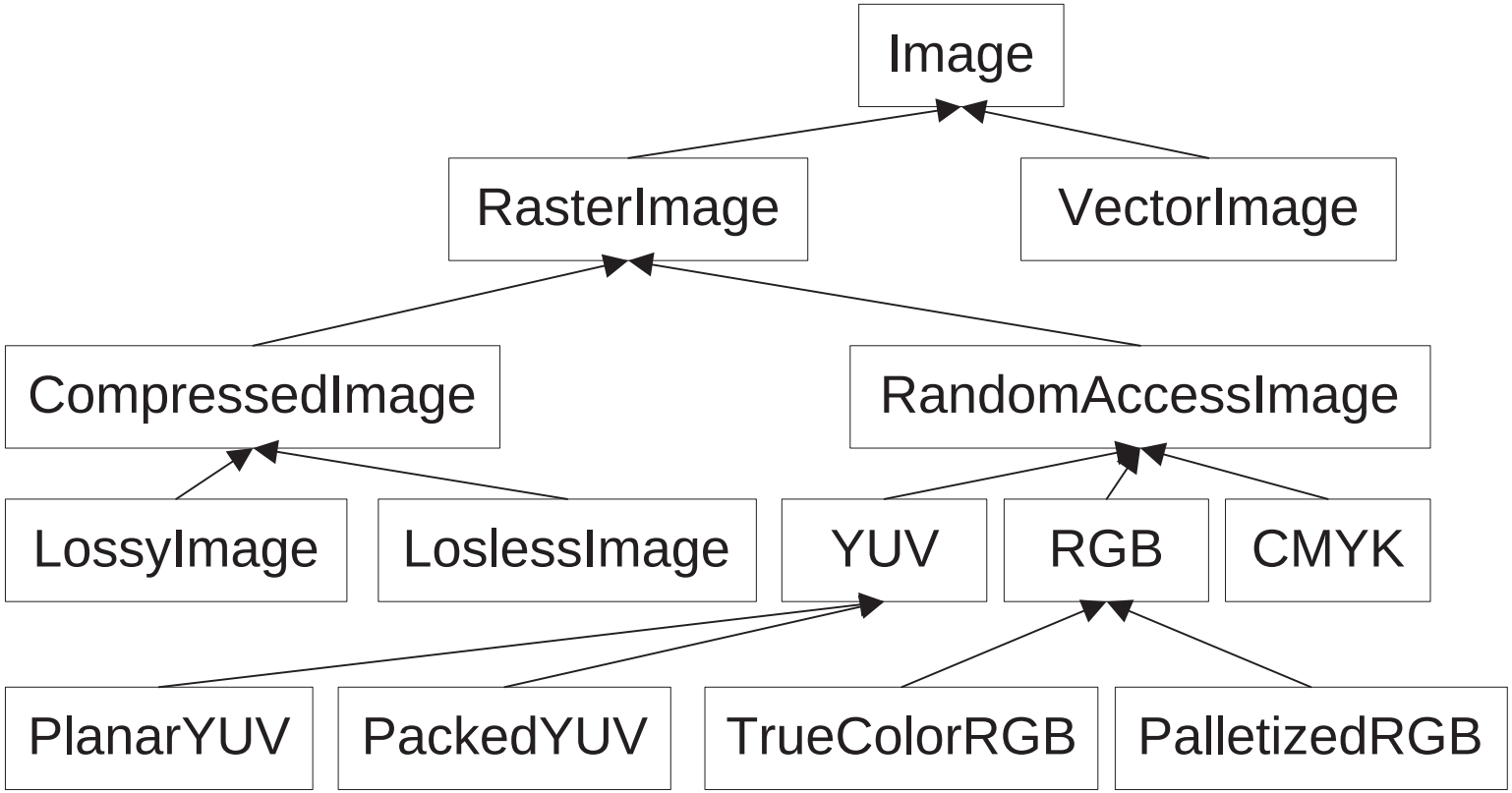
Parasol
Smarter computing.
Texas A&M University

Application Domains

- Expression Problem
  - Image Format Conversions
  - Algorithm selection according to dynamic properties of objects
  - Action System
- Binary Method Problem
  - Shape intersection
  - Object comparison
  - Operations in dynamically typed languages

Parasol
Smarter computing.
Texas A&M University

Application Domain: Image Format Conversion



Parasol
Smarter computing.
Texas A&M University

Application Domain: Image Format Conversion (2)

			Non-palletized RGB				Interleaved YUV				Planar YUV				
		RGB8	RGB555	RGB565	RGB24	RGB32	YUY2	YVYU	UYVY	Y41P	CLJR	I420	YV12	YVU9	Y800
	RGB8		U	U	U	U	CC	CC	CC	CC	CC	CC	CC	CC	CP
Non-palletized RGB	RGB55	D		U	U	U	CC	CC	CC	CC	CC	CC	CC	CC	GS
	RGB56	D	D		U	U	CC	CC	CC	CC	CC	CC	CC	CC	GS
	RGB24	D	D	D		+ 0	CC	CC	CC	CC	CC	CC	CC	CC	GS
	RGB32	D	D	D	- 0		CC	CC	CC	CC	CC	CC	CC	CC	GS
Interleaved YUV	YUY2	CC	CC	CC	CC	CC		VU	2B	D	D	DR	DR	DR	RY
	YVYU	CC	CC	CC	CC	CC	VU		4B	D	D	DR	DR	DR	RY
	UYVY	CC	CC	CC	CC	CC	2B	4B		21	D	DR	DR	DR	RY
	Y41P	CC	CC	CC	CC	CC	U	U	12		D	RR	RR	DR	RY
	CLJR	CC	CC	CC	CC	CC	U	U	U	U		UR	UR	UR	UY
Planar YUV	I420	CC	CC	CC	CC	CC	UR	UR	UR	RR	DR		uv	YD	CY
	YV12	CC	CC	CC	CC	CC	UR	UR	UR	RR	DR	uv		YD	CY
	YVU9	CC	CC	CC	CC	CC	UR	UR	UR	UR	DR	YU	YU		CY
	Y800	AP	2	2	3	30	Y0	Y0	Y0	Y0	DY	YZ	YZ	YZ	

Double Dispatch/Visitor Pattern

```
struct Shape {
  virtual bool intersect(Shape&);
  virtual bool intersect(Circle&);

  virtual bool accept(Shape& obj2) { return obj2.intersect(*this); }
};

struct Circle : Shape {
  virtual bool intersect(Shape&);
  virtual bool intersect(Circle&);

  virtual bool accept(Shape& obj2) { return obj2.intersect(*this); }
};

bool shape_intersect(Shape& obj1, Shape& obj2) {
  return obj1.accept(obj2);
}
```

Parasol
Smarter computing.
Texas A&M University

Ambiguities

for example:

```
struct Shape {};
struct Circle : Shape {};

void intersect(virtual Shape&, virtual Shape&);
void intersect(virtual Circle&, virtual Shape&);
void intersect(virtual Shape&, virtual Circle&);
```

Parasol
Smarter computing.
Texas A&M University

Types of Ambiguities

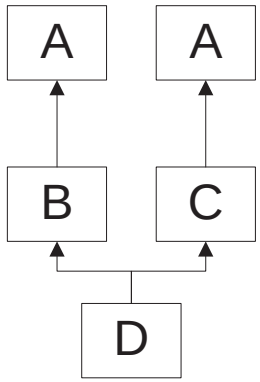
- Resolvable Ambiguities
  - Detected before progam invocation
  - Classes are available
- Late Ambiguities
  - e.g.: dynamic linking, local classes

Parasol
Smarter computing.
Texas A&M University

Ambiguities: Virtual vs overloaded functions

- Are the calls to foo ambiguous?
  - Overload resolution
  - Virtual function call
  - Open Multimethods

```
void foo(A&);
void foo(B&);

void bar(A& a) { foo(a); }
void bar(D& d) { foo(d); } // ok
```

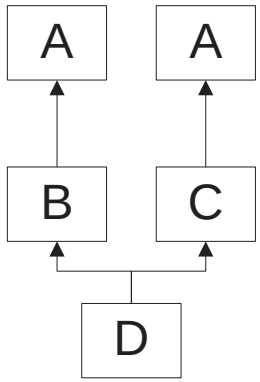


Parasol
Smarter computing.
Texas A&M University

Ambiguities: Virtual vs overloaded functions

- Are the calls to foo ambiguous?
  - Overload resolution
  - Virtual function call
  - Open Multimethods

```
struct A { virtual void foo(); };
struct B : A { virtual void foo(); };

void bar(A& a) { a.foo(); }
void bar(D& d) { d.foo(); } // compiler error
```

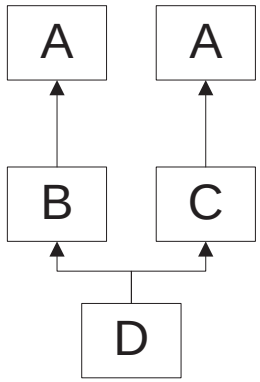


Parasol
Smarter computing.
Texas A&M University

Ambiguities: Virtual vs overloaded functions

- Are the calls to foo ambiguous?
  - Overload resolution
  - Virtual function call
  - Open Multimethods

```
void foo(virtual A&);
void foo(virtual B&);

void bar(A& a) { foo(a); }
void bar(D& d) { foo(d); } // compiler error
```

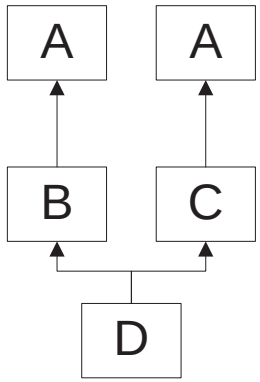


Parasol
Smarter computing.
Texas A&M University

Ambiguities: Virtual vs overloaded functions

- Are the calls to foo ambiguous?
  - Overload resolution
  - Virtual function call
  - Open Multimethods

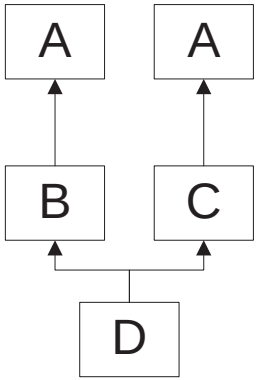
```
void foo(virtual A&);
void foo(virtual B&);

void bar(A& a) { foo(a); }
void bar(D& d) { foo(static_cast<B&>(d)); }
```



Parasol
Smarter computing.
Texas A&M University

Ambiguities: Repeated Inheritance

void foo(**virtual** A&, **virtual** A&);
void foo(**virtual** B&, **virtual** C&);
void foo(**virtual** C&, **virtual** B&);
void foo(**virtual** B&, **virtual** D&);

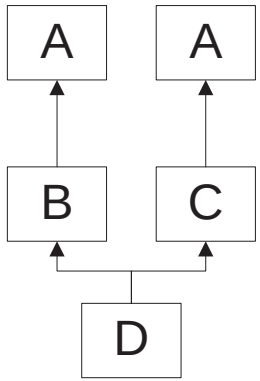


$1^{st} \backslash 2^{nd}$	A	B	C	D_B	D_C
A	AA	AA	AA	AA	AA
B	AA	AA	BC	BD	BD
C	AA	CB	AA	CB	AA
D_B	AA	AA	BC	BD	BD
D_C	AA	CB	AA	CB	AA

Parasol
Smarter computing.
Texas A&M University

Ambiguities: Repeated Inheritance

void foo(**virtual** A&, **virtual** A&);
void foo(**virtual** B&, **virtual** C&);
void foo(**virtual** C&, **virtual** B&);
void foo(**virtual** B&, **virtual** D&);

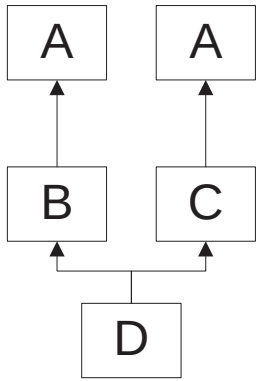


$1^{st} \backslash 2^{nd}$	A	B	C	D_B	D_C
A	AA	AA	AA	AA	AA
B	AA	AA	BC	BD	BD
C	AA	CB	AA	*CB*	*AA*
D_B	AA	*AA*	*BC*	*BD*	*BD*
D_C	AA	*CB*	*AA*	*CB*	*AA*

Parasol
Smarter computing.
Texas A&M University

Ambiguities: Repeated Inheritance

void foo(**virtual** A&, **virtual** A&);
void foo(**virtual** B&, **virtual** C&);
void foo(**virtual** C&, **virtual** B&);
void foo(**virtual** B&, **virtual** D&);

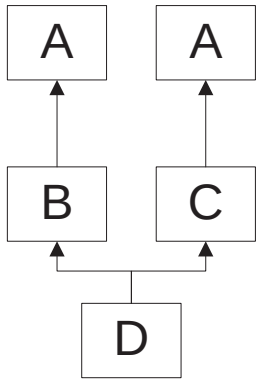


$1^{st} \backslash 2^{nd}$	A	B	C	D_B	D_C
A	AA	AA	AA	AA	AA
B	AA	AA	BC	BD	BD
C	AA	CB	AA	*CB*	*AA*
D_B	AA	*AA*	*BC*	*BD*	*BD*
D_C	AA	*CB*	*AA*	*CB*	*AA*

Parasol
Smarter computing.
Texas A&M University

Ambiguities: Repeated Inheritance

void foo(**virtual** A&, **virtual** A&);
void foo(**virtual** B&, **virtual** C&);
void foo(**virtual** C&, **virtual** B&);
void foo(**virtual** B&, **virtual** D&);

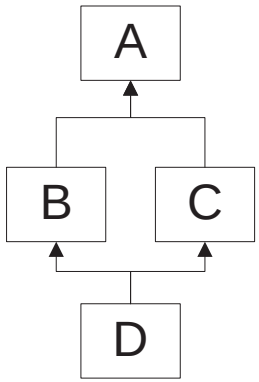


$1^{st} \backslash 2^{nd}$	A	B	C	D_B	D_C
A	AA	AA	AA	AA	AA
B	AA	AA	BC	BD	BD
C	AA	CB	AA	*CB*	*AA*
D_B	AA	*AA*	*BC*	*BD*	*BD*
D_C	AA	*CB*	*AA*	*CB*	*AA*

Parasol
Smarter computing.
Texas A&M University

Ambiguity Resolution: Virtual Inheritance

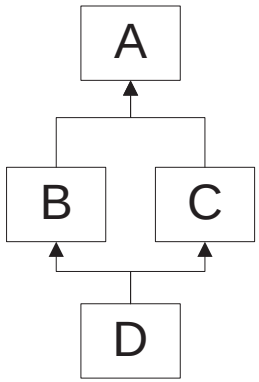
```
void foo(virtual A&);
void foo(virtual B&);
void foo(virtual C&);

void bar(A& a) { foo(a); }

bar(d);
```



Parasol
Smarter computing.
Texas A&M University

Ambiguity Resolution: Virtual Inheritance

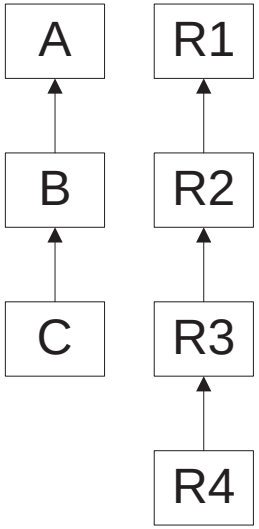
```
void foo(virtual A&);
void foo(virtual B&);
void foo(virtual C&);

void bar(A& a) { foo(a); }

bar(d);

void foo(virtual D&);
```



Parasol
Smarter computing.
Texas A&M University

Covariant Return Type - Ambiguity Resolution

R1∗ foo(**virtual** A&, **virtual** A&);
R2∗ foo(**virtual** A&, **virtual** B&);
R3∗ foo(**virtual** B&, **virtual** A&);
R4∗ foo(**virtual** B&, **virtual** C&);

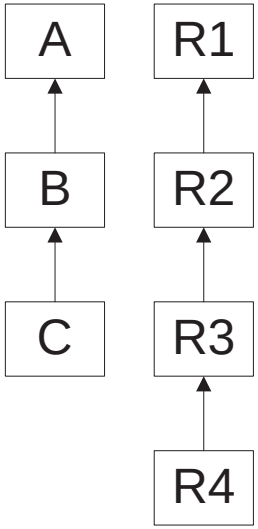


$1^{st} \backslash 2^{nd}$	A	B	C
A	AA	AB	AB
B	BA		BC
C	BA	BA	

Parasol
Smarter computing.
Texas A&M University

Covariant Return Type - Ambiguity Resolution

R1∗ foo(**virtual** A&, **virtual** A&);
R2∗ foo(**virtual** A&, **virtual** B&);
R3∗ foo(**virtual** B&, **virtual** A&);
R4∗ foo(**virtual** B&, **virtual** C&);

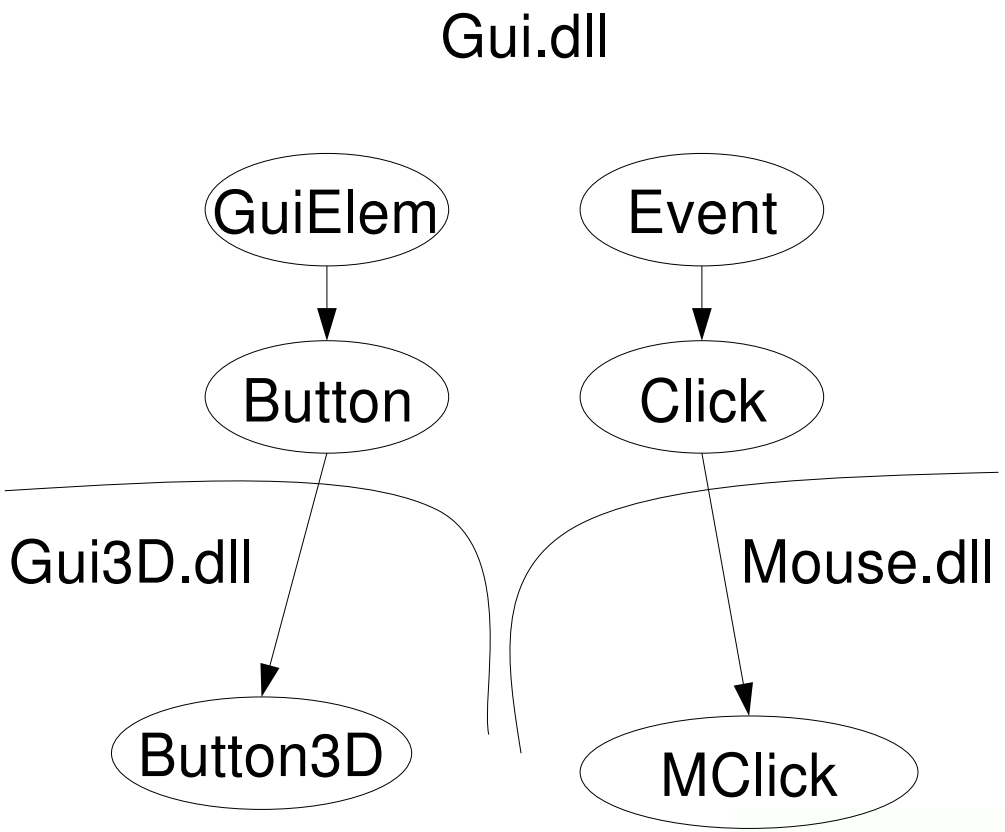


$1^{st} \backslash 2^{nd}$	A	B	C
A	AA	AB	AB
B	BA	BA	BC
C	BA	BA	BC

Parasol
Smarter computing.
Texas A&M University

Late Ambiguities - The Problem

```
// Gui.dll
void handle(GuiElem&, Event&);
void handle(Button&, Click&);

// Gui3D.dll
void handle(Button3D&, Click&);

// Mouse.dll
void handle(Button&, MClick&);

handle(button3d, mclick);

// oops, problem
```



Parasol
Smarter computing.
Texas A&M University

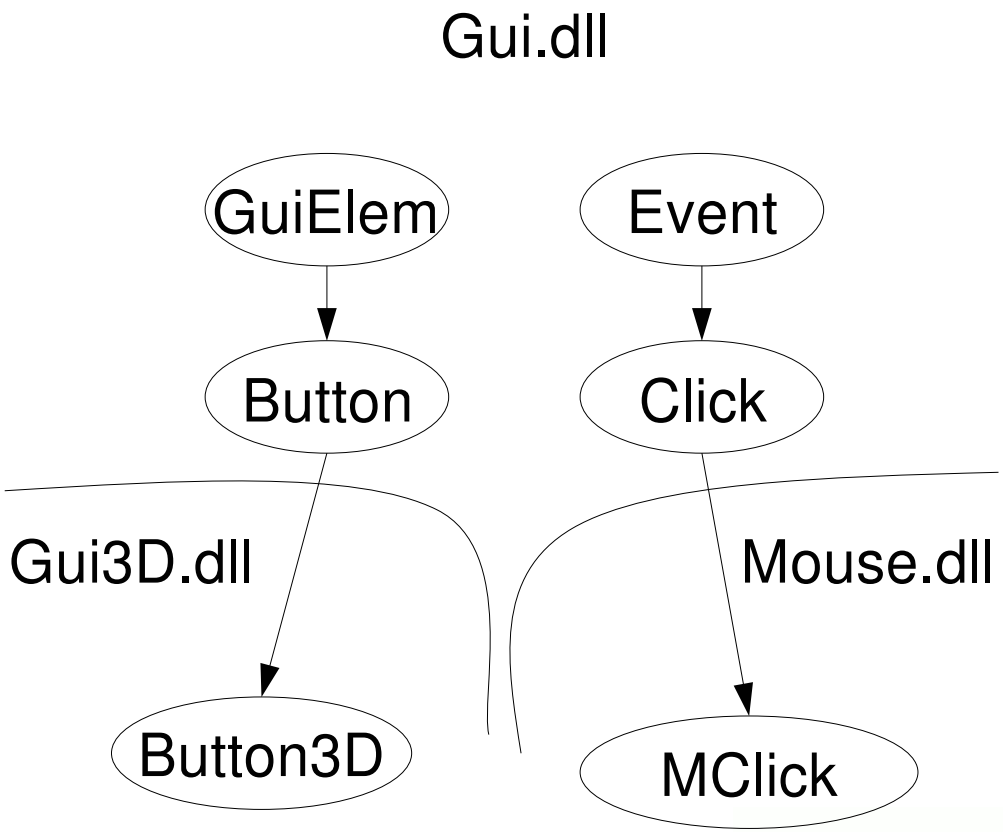
Ambiguities: Related Work

- CLOS (Steele)
  - asymmetric dispatch
- Cecil (Chambers)
  - whole program view
- MultiJava (Clifton et al.)
  - single inheritance
- Relaxed MultiJava (Millstein et al.):
  - programmmer resolved through *Glue - methods*.

Parasol
Smarter computing.
Texas A&M University

Late Ambiguities - A Solution

```
// Gui.dll
void handle(GuiElem&, Event&);
void handle(Button&, Click&);

// Gui3D.dll
void handle(Button3D&, Click&);

// Mouse.dll
void handle(Button&, MClick&);

handle(button3d, mclick);

// either Mouse.dll or Gui3D.dll
should be fine
```



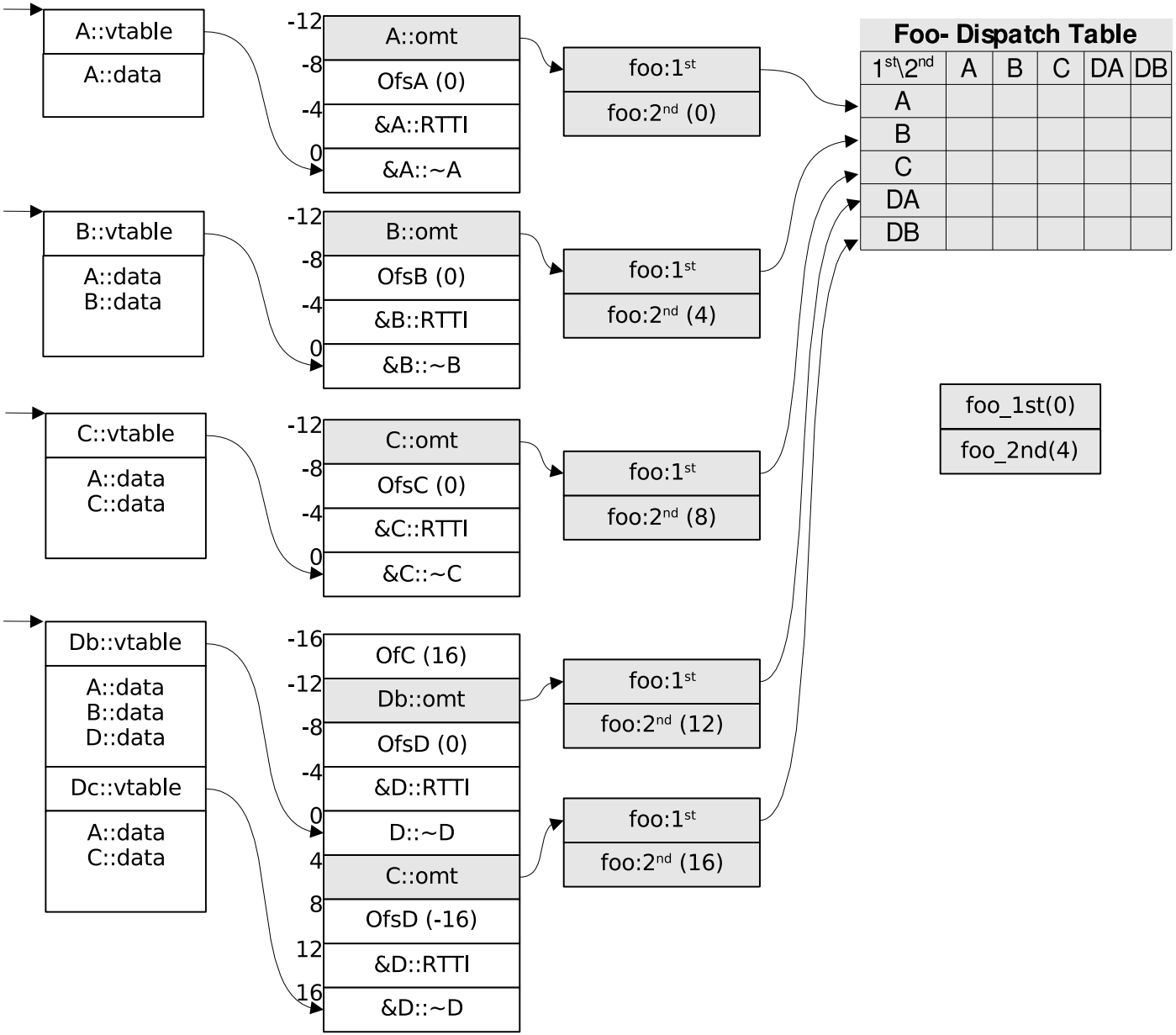
Parasol
Smarter computing.
Texas A&M University

Late Ambiguities - A Solution (2)

- Resolves late ambiguities to any best overrider
  - deterministic
  - but non specfied
- Conformant with Liskov's substitution principle

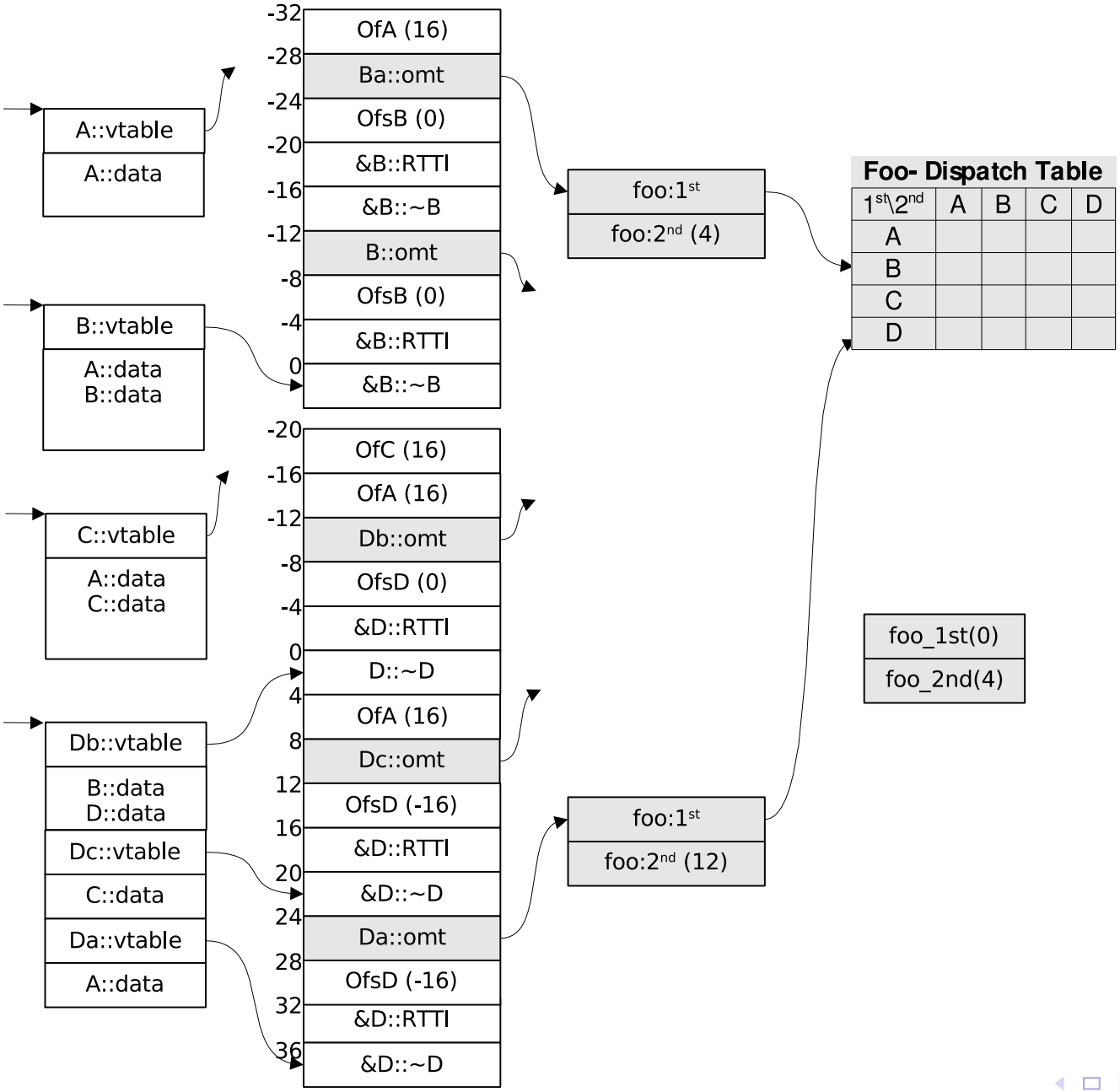
Parasol
Smarter computing.
Texas A&M University

Object Model (Repeated Inheritance)



Parasol
Smarter computing.
Texas A&M University

Object Model (Virtual Inheritance)



Parasol
Smarter computing.
Texas A&M University

Experiment

Shape intersection

- 20 shape classes
- 400 dispatch combination
- 40 implemented intersect functions

Parasol
Smarter computing.
Texas A&M University

Results

Approach	Cycles/Loop Pentium-D	Cycles/Loop Core2Duo	Size (bytes) Linux
Virtual function	75	55	n/a
C++ Multi-method	78	60	19 547
C++ Open-method	82	63	19 725
EDG/Omm	82	64	n/a
Double Cpp	120	82	20 859
C++ Visitor	132	82	35 289
Chinese Remainders	175	103	n/a
Cmm (constant time)	415	239	112 250
Cmm	1 320	772	111 305
Loki Library	3 670	2 238	34 908

Parasol
Smarter computing.
Texas A&M University

- intersect of 20 shape classes

Conclusion

- generalized double dispatch for the *C++ object model*
- introduced *co-variant return type* for ambiguity resolution
- extends the *C++ compilation and IA-64 object model*
- proposed novel idea for *ambiguity resolution for DLLs*

Parasol
Smarter computing.
Texas A&M University

Future Work

Outlook

- Templated virtual functions
- Pointers to Open-methods
- Allow calling base-methods (similar to base-class calls)
- Space Optimizations

Parasol
Smarter computing.
Texas A&M University

Thank You!