Overloaded Methods in Babel

Tammy Dahlgren, Tom Epperly, Scott Kohn, and Gary Kumfert

Center for Applied Scientific Computing

Common Component Architecture Working Group

October 3-4, 2002

Agenda

- **Background**
- **SIDL Example**
- **Generated Impl Names**
- **Generated Stub Names**
- **More information**

What is method overloading?

Method overloading is the object-oriented practice of defining **more than one method** in a class with the **same name** *but different signatures*.

Why is it useful?

Method overloading allows convenient **reuse of a method name** when the underlying implementation varies based on the arguments.

As an example, below is a snippet of SIDL associated with a simulation.

```
class Simulation {
    void init (in InputDatabase database);         Right!
    void init[FromRestart] (in string dir, in int num,
                            in InputDatabase database);
```

Short name

...

Name extension

Wrong!

```
    void write[Restart] (in string filename, in int sequence_ext);
    void write[Vis] (in string filename, in int sequence);
}
```

Simulation.sidl
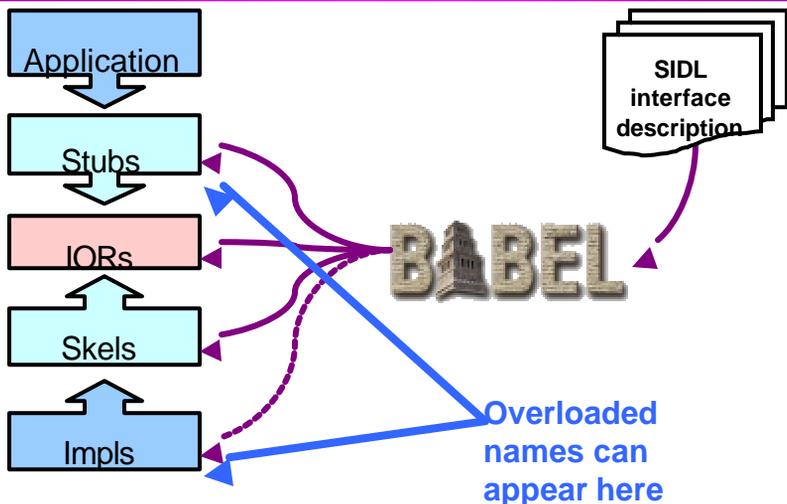
Based upon an exact match of arguments – type(s) and ordering!

Glue code can then be generated for each supported language from the SIDL file.

Application

Stubs

IORs

Skels

Impls

SIDL interface description

BABEL

Overloaded names can appear here

Generated Stub method names used by the caller depend upon language support.

Lang.	init	init[FromRestart]
C	Simulation_init	Simulation_initFromRestart
C++	init	init
Python	init	initFromRestart
F77/F90	Simulation_init_f	Simulation_initFromRestart_f
Java	init	init

Similarly, Impl method names depend upon language support.

Lang.	init	init[FromRestart]
C	impl_Simulation_init	impl_Simulation_initFromRestart
C++	init	init
Python	init	initFromRestart
F77/F90	Simulation_init_impl	Simulation_initFromRestart_impl

For more information, refer to the documentation and examples.

- **Babel User's Guide**
  - —Contains an example SIDL file (in section 3.4.4) that is used by the language binding chapters to illustrate how each language calls the methods.

- **Examples**
  - —The *regression* tests' *overload* subdirectory contains code for each of the supported languages.