

Übungsaufgabe #4: Automatische Erzeugung von Stubs

21.05.2003

In der Aufgabe 3 wurden manuell Stubs für Client und Server für ein bestimmtes Objekt erstellt. In dieser Aufgabe soll diese automatisiert werden, ausgehend von einer Schnittstellenbeschreibung. Zuvor soll jedoch das Marshalling um einige Funktionen erweitert werden.

a) Erweiterung des Marshallings

In dieser Teilaufgabe soll das Marshalling um eine Unterstützung für Arrays von primitiven Datentypen und für eine Call-By-Value-Result-Semantik erweitert werden.

Arrays mit fester Länge: Für jeden primitiven Datentyp ist in der Klasse `Message` eine Methode `bool writeArray(int len, type data[])` und eine entsprechende `readArray`-Methode vorzusehen. Es kann davon ausgegangen werden, dass `readArray/writeArray` von den Stubs auf Client- und Serverseite mit gleicher Längenangabe aufgerufen werden. Beim `readArray` hat der Aufrufer einen hinreichend grossen Puffer bereitzustellen.

Call by Value-Result: Bisher werden Methodenparameter mit *Call-by-Value* Semantik zum Server übertragen. Um etwas ähnliches zu *Call-by-Referenz* bereitzustellen, lässt sich eine *Call-by-Value-Result* Semantik implementieren. Hierzu ist ein Parameter zunächst wie bei *Call-by-Value* zum Server zu übertragen. Bei Rückkehr aus dem Funktionsaufruf ist der (möglicherweise vom Server modifizierte) Parameter wieder zurück zum Klienten zu übertragen.

Erstellen Sie auch hierzu einen kleinen Testserver mit entsprechend angepassten Stubs. Parameter, die per Value-Result übertragen werden sollen, sind als Referenzparameter zu deklarieren (Bsp. `int16_t ¶m` statt `int16_t param`).

b) automatische Erzeugung von Stubs und Skeletons

In der Übung haben Sie das Tool *IDLflex* zur automatischen Erzeugung von Stubs und Skeletons kennengelernt. Dieses sollen Sie als Basis einsetzen, um automatisch Stubs für Ihr RPC-System zu erstellen. Hierzu ist eine entsprechende XML-Steuerungsdatei für *IDLflex* zu erstellen. Als Eingabe soll eine Schnittstellenbeschreibung in CORBA IDL dienen.

In CORBA IDL kann man für jeden Parameter eine Richtung angeben, in die er transportiert werden soll (in/out/inout). Diese Angabe soll durch die erzeugten Stubs (und Skeletons) unterstützt werden. "in" ist dabei auf *Call-by-Value* abzubilden, "inout" auf *Call-by-Value-Result*. "out" lässt sich wie "inout" behandeln, die Übertragung der Daten zum Server kann dabei aber unterdrückt werden.

Alle bisher manuell erzeugten Stubs (Teilaufgabe (a) sowie vorhergegangene Aufgabe 3) sollten sich damit nun auch automatisch generieren lassen.

Abgabe: bis 27.05.2003 12:00 Uhr (eine Woche Bearbeitungszeit!)

Abgabe mittels `/proj/i4vs/pub/abgabe` (Abgabe in 2er oder 3er Gruppen möglich)