

Übungsaufgabe #4: Automatische Erzeugung von Stubs

11.06.2007

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

a) Erweiterung des Marshallings um Unterstützung für Arrays primitiver Datentypen

(nur für Informatiker)

In dieser Teilaufgabe soll das Marshalling um eine Unterstützung für Arrays fester Länge erweitert werden. 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.

b) automatische Erzeugung von Stubs und Skeletons

In der Übung haben Sie das Tool *IDLflex* zur automatischen Erzeugung von Stubs und Skeletons kennen gelernt. 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 dient eine Schnittstellenbeschreibung in CORBA IDL.

Alle bisher manuell erzeugten Stubs (Teilaufgabe **a** (sofern bearbeitet) sowie vorhergegangene Aufgabe 3) sollten sich damit nun auch automatisch generieren lassen. Es müssen (zunächst) nur "in"-Parameter unterstützt werden.

c) Call-by-value/result

(nur für Informatiker)

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 Client zu übertragen.

Erstellen Sie hierzu zunächst einen kleinen Testserver mit entsprechend manuell angepassten Stubs. Parameter, die per *value/result* übertragen werden, sind als Referenzparameter zu deklarieren (Bsp. `int16_t ¶m` statt `int16_t param`).

Erweitern Sie dann die *IDLflex*-Mappingbeschreibung für *Call-by-value/result*. 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" auf *call-by-result*.

Abgabe: bis 03.07.2007/12:00 Uhr

Alle benötigten Dateien sind im Verzeichnis *fax* des SVN-Repositories abzulegen.

Übungen zu Verteilte Systeme