An IDL to SDL2000 compiler

M. de Cabo, M. Rodríguez, E. Fernández
Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática
E.T.S.I. de Telecomunicación. Universidad de Valladolid.
{*marcab,manrod,eusebio*}*@gbien.tel.uva.es*

Abstract

Distributed systems evolution has led telecommunication management network (TMN) systems to use object-oriented middleware paradigm, mainly CORBA (Common Object Request Broker Architecture). CORBA only includes a mechanism for defining object interfaces not for specifying the behaviour of these objects. The behaviour of a TMN system has to follow concrete and restrictive specifications, which are not reflected in CORBA IDL, at least not in a clear and unambiguously way. It is in this situation where the advantage of using an FDT (Formal Description Technique) like SDL (Specification and Description Language) appears.

This work is the first step towards the overall goal of formally specifying the behaviour of the objects implementing management IDL interfaces.

1 IDL to SDL Mapping

Formalising an IDL to SDL mapping is the first requirement because there is not a standard IDL-SDL mapping. An IDL to SDL compiler was developed in order to automate this translation. It translates IDL specifications into SDL structures that have to be completed with corresponding finite-state machines to obtain the desired behaviour.

In [1] some guidelines to translate IDL elements into SDL96, [2], constructions are given. Guidelines related to predefined types, synchronous and asynchronous methods have been used. New rules based on SDL2000 [3] new characteristics (exceptions, interfaces, direct use of ASN.1 concepts like CHOICE or OPTIONAL) have been developed. And some ideas from other IDL mappings have been adopted for those SDL types that are not inmediately translatable.

IDL to SDL mapping gives us the possibility to obtain SDL skeletons corresponding to a given IDL definition. Applying this mapping to generic management interfaces from [4] it will be obtained a generic SDL skeleton. This can be used as a base for SDL systems of specific management interfaces developed using the guidelines in [4]. Next step towards the overall objective is to define the behaviour of these skeletons.

2 Future Work

The SDL skeletons have to be filled in with behaviour. The behaviour will be implemented by means of generic processes, procedures and finite-state machines, representing generic managed object structures and behaviour. They could be made specific by means of SDL specialisation. Such generic elements will form the basis for the behaviour specification of management systems like generic IDL interfaces are the basis for defining concrete interfaces.

Once the SDL system is completed, existing SDL2000 tools can generate code from the specification to obtain a CORBA application. That application will offer the initial IDL interfaces to other CORBA objects and will behave as we specify.

The work is still in progress. Our first step, the compiler, is being optimised. The second, the generation of general behaviour, is being developed using the comments of [4] and the work developed in [5].

References

[1] Telelogic AB. *Telelogic Tau 3.6 User's Manual: the CORBA integration*, September 1999.

[2] ITU-T Recommendation Z.100. Addendum 1 SDL: Specification and Description Language, October 1996.

[3] ITU-T. *ITU-T Recommendation Z.100(11/99). SDL: Specification and Description Language*, November 1999.

[4] ITU-T. *ITU-T X.780 (2001): TMN Guidelines for Defining CORBA Managed Objects*, January 2001.

[5] M. Rodríguez. *Contribución a la especificación formal de sistemas TMN a partir del modelo de información de gestión.* PhD thesis, E.T.S.I.T Universidad de Valladolid, November 1999.