OUN:

A Formalism for Open, Object Oriented, Distributed Systems

Olaf Owe Isabelle Ryl

Department of Informatics
University of Oslo
{olaf,isabelle}@ifi.uio.no

OUN is a notation developed within the frame of the ADAPT-FT project. The goal of this project is to provide a platform supporting the formal development of object oriented open distributed systems. The purpose of this platform is to support formal reasoning and, in contrast to most formal methods, deal with the main aspects of open distributed systems such as mobility, flexibility, and reflection. In the platform the notation will be used combined with UML, but is not dedicated to it.

The ultimate purpose of the platform is to enable development of systems satisfying given safety requirements. Safety reasoning is difficult unless the constructs of the specification and design language are chosen carefully. We therefore introduce our own language, OUN, for system specification and design. Openness is taken into account by allowing dynamic addition of (sub)interfaces and (sub)classes, as well as dynamic extension of classes. And as usual in object oriented languages, objects may be created dynamically and their identities may be communicated. To facilitate reasoning, we insist on static typing and static correctness proofs. This leads to some language restrictions, compared to Corba and Java RMI.

OUN is a trace-based notation which allows us to specify contracts, interfaces and classes using high level object oriented concepts like multiple inheritance of interfaces and classes, and allowing implementation of several interfaces by a class. Interface specifications characterize observable behavior of one object, contracts restrict the behavior of a subsystem with two or more objects, while class specifications characterize local attributes as well as internal and external behavior. The notation will take account of synchronous as well as asynchronous communications.

It is essential that the notation is practically useful, and not only reserved for specialists with strong mathematical background: The basic concepts are easy-to-understand, the formulas are expressed in first order logic and the syntax is easy-to-use.