Ph.D. thesis of Ciaran McHale

Synchronisation in Concurrent, Object-oriented Languages: Expressive Power, Genericity and Inheritance

Abstract

This thesis explores synchronisation in concurrent, object-oriented languages (COOPLs) and makes contributions in the following three areas.

A paradigm for the development of synchronisation mechanisms.

This thesis defines a paradigm, called *service-object synchronisation* (SOS), for the development of synchronisation mechanisms. I show that SOS-based mechanisms have several desirable characteristics: (i) excellent expressive power without complexity; (ii) a complete separation of synchronisation code and sequential code; and (iii) they avoid unsafe access to the instance variables of an object. Furthermore, the prototype implementation of a sample, SOS-based mechanism shows that the concepts of the SOS paradigm can be easily introduced to an object-oriented language.

Generic synchronisation policies.

Several different synchronisation policies occur frequently in practice. It is time-consuming and error-prone to have to re-implement such policies repeatedly in different classes that use them. It would be preferable to write *generic* synchronisation policies that could then be instantiated upon the operations of different class as desired. For example, consider a class that contains three operations, *A*, *B* and *C*, that examine some instance variables, and two other operations, *D* and *E*, that update instance variables. A suitable policy for this class might be as follows:

ReadersWriter[{A, B, C}, {D, E}]

This readers/writer policy is instantiated upon two sets of operations. The first, "{A, B, C}", is a set of read-style operations and the second a set of write-style operations.

In this thesis, I show that the SOS paradigm makes it possible to provide language support for generic synchronisation policies; and that this support surpasses that provided by previous researchers.

The most obvious benefit of generic synchronisation policies is code reuse. However, I demonstrate that generic synchronisation policies offer additional benefits, including facilitating the optimisation of synchronisation code.

Analysis of the problems with the use of inheritance in COOPLs

It is well-known that there are problems associated with the use of inheritance in COOPLs. Most of the research to date in this area has assumed that one of these problems is due to a conflict between synchronisation and inheritance. I show that this assumption is incorrect and that the problem is rooted in the conflicting interaction of two different uses of inheritance. I survey the conflict in a number of mechanisms for inheriting synchronisation code and show that the use of generic synchronisation policies can drastically reduce the harmful effects of the conflict.

Citation details

Citation details (in BibTeX format) for my thesis are given below.

```
@phdthesis{mchale-94,
  author  = "Ciaran McHale",
  title   = "Synchronisation in Concurrent,
             Object-oriented Languages:
             Expressive Power, Genericity
             and Inheritance",
  year    = "1994",
  month   = oct,
  school  = "Department of Computer Science,
             Trinity College",
  address = "Dublin 2, Ireland",
  PDF     = "http://CiaranMcHale.com/download/phd-thesis.pdf",
}
```