integrating
Fortran
and
XML

Participant documentation

Toby White and Andrew Walker
Department of Earth Sciences
University of Cambridge

28th - 30th January 2008

e-Science Institute

NIEeS

Contents

Timetable and course outline

Monday 28th January

12:00 - 14:00	Registration and lunch at eSI	
14:00 - 14:15	Brief introduction to us, eSI, the course and facilities	
14:15 - 15:00	**Lecture 1:** Introduction to the course and overview of XML	Toby White
15:00 - 15:30	Writing XML, introducing the FoX library	Andrew Walker
15:30 - 16:00	Tea	
16:00 - 17:15	**Practical 1:** Using FoX to write HTML and MathML	
17:15 - 17:30	Review	

Tuesday 29th January

09:00 - 10:00	**Lecture 2:** XML languages	Toby White
10:00 - 10:30	Introduction to KML, Google Earth and practical 2.	Andrew Walker
10:30 - 11:00	Coffee	
11:00 - 12:00	**Practical 2:** Adding XML output to real Fortran code.	
12:00 - 14:00	Lunch	
14:00 - 15:00	**Lecture 3:** The XML landscape	Toby White
15:00 - 15:45	Introduction to scripting with XPath and an introduction to Practical 3	Andrew Walker
15:45 - 16:15	Tea	
16:15 - 17:30	**Practical 3:** Using XPath to extract data from XML documents	

Wednesday 30th January

09:00 - 10:00	Lecture 4: Accessing XML data from Fortran: SAX, DOM and Beyond	Toby White
10:00 - 10:30	Introduction to practical 4	Toby White
10:30 - 11:00	Coffee	
11:00 - 12:15	**Practical 4:** Reading XML data into a Scientific Application	
12:15 - 12:45	Course review	

Welcome to iFaX

iFaX - integrating Fortran and XML

In recent years, a number of changes have come about in the way that data is used, fuelled by a combination of increasing computer speed, and increasing network speed.

There is a growing problem of information management - computers are so fast now that we can run useful calculations, producing megabytes of output, even on our own desktop computers. In fact, we are capable of creating and collating such enormous amounts of data that we are in danger of losing our ability to sensibly manage it - it is becoming increasingly difficult to browse, search, and analyse our own data.

There is also a growing desire to be able to share data more effectively - the speed of data transfer across the internet has made it much easier to send raw data to collaborators, or even publish it to the world at large. However, it is rarely easy to immediately use somebody else's data, with no knowledge of provenance, or context, or even common data formats.

Furthermore, driven both by the increasing speed of desktop computers (and thus their ability to display high-quality graphics) and the increasing speed of the network (enabling vast amounts of data to be transferred around, thus allowing interactive, remotely hosted, applications), there is a flourishing growth in exciting and innovative visualization technologies. SVG and Google Earth are cases in point.

What all of these issues have in common is XML. XML is a large part of the answer to any data management solution, enabling storage of metadata, and enabling you to encode data in commonly understood, well-documented data formats. In addition, XML has become the language of choice for data transfer to visualization applications like Google Earth.

However, in the computational scientific world, most of our data is produced by programs written in Fortran. Fortran and XML do not naturally fit together, and there are a number of issues to be overcome in bringing a Fortran workflow into an XML environment.

In this workshop, we have drawn on the combined expertise of a number of scientists, who over the past few years, under the auspices of the UK eScience programme, have successfully dealt with XML in a scientific context, and bridged the Fortran-XML gap.

We will teach you about XML, its background, and its uses in a scientific context.

We will introduce you to a Fortran library which will let you write and read XML directly from a Fortran program, without any intermediaries.

We will teach you how to write analysis scripts to quickly and easily extract data from XML, much like you would write `grep` or `Perl` scripts to do so from a text file.

And throughout the course, we will be working with real Fortran scientific software, to show you exactly how you bring existing Fortran codes into an XML-aware environment.

In short, we will bring you up to speed with the skills necessary for operating as a computational scientist in the world of XML.