Software Engineering in the Computational Science and Engineering Department

David J Worth and Chris Greenough
Thursday 23rd March 2006

Software Engineering Group
Computational Science & Engineering Department
Rutherford Appleton Laboratory

d.j.worth@rl.ac.uk
c.greenough@rl.ac.uk
Outline

- CSED Activities
  - http://www.cse.clrc.ac.uk/index.shtml

- Software Engineering in CSED

- Legacy Software
The Computational Science and Engineering Department

- Based at DL and RAL

- Development and application of powerful simulation codes
  - Usually collaborating with Universities
  - Emphasis on high performance
    - Algorithms
    - Hardware – HPCx (http://www.hpcx.ac.uk/), HP cluster coming soon

- Interests
  - Quantum chemistry,
  - Molecular simulation,
  - Materials simulation,
  - Engineering

- Collaborative Computational Projects (CCPs) http://www.ccp.ac.uk/
Materials Science and Condensed Matter

Modelling active site catalysts using combined QM/MM methods to design more specific, more environmentally friendly, more active systems capable of working at lower temperatures and pressures.

Slow diffusion!
Bluemoon method
Fixed and flexible framework
Reaction path found
Free energy profiles
MC method for $D_0$

(5x5) $\text{Cr}_2\text{O}_3$ surface containing 700 atoms (14850 basis functions) per unit cell. Such calculations are essential in order to study the role of defects in determining the properties of real materials.

Accurate modelling of molecular interactions for crystal structure prediction.
Life Sciences

Modelling active sites of enzymes in solvents at room temperature - simulation of TIM at 300K to study mechanism of active site

Simulations of liposomes coating DNA strands prior to transport across cell membranes

Virtual Outer Membrane - molecular dynamics simulations of transport channels through membranes require 2 million atoms to be modelled for 100 ns with multiple comparative runs to generate statistics (Mark Sansom)
Computational Engineering

Modelling complex geometries

Modelling turbulence (Neil Sandham)

Microfluidics

Diffusion of red blood cells
\[ D = 6.8 \times 10^{-10} \text{ cm}^2/\text{s} \]

Diffusion of serum albumin
\[ D = 6.5 \times 10^{-7} \text{ cm}^2/\text{s} \]
Ocean and Climate Models

- A 1/12° Ocean Model
  - has 608 million grid cells
  - needs 60 Gbyte storage
  - needs $40 \times 10^{15}$ floating point operations/model year
  - produces a 20 Gbyte data set every 3 model days

- A comparative climate model with ocean, atmosphere and land sub-models needs about twice the resources.

- Greenhouse effect, raised CO$_2$ emissions, ozone depletion, storm and gulf stream variability, regional shelf edge models, biological sub-models
Numerical Analysis Group

Fields of Interest
- Sparse linear solvers
- Nonlinear optimization
- Numerical algebra and PDEs

Major Projects
- EPSRC grant
- HSL - collection of ISO Fortran codes for large scale scientific computation
- Galahad - thread-safe library of Fortran 90 packages for large-scale nonlinear optimization
Software Engineering Group

- **Software Engineering Support Programme (SESP)**
  - Funded by EPSRC
  - SLA for CSED and CCPs

- **Intelligent Agent Technology**
  - University of Sheffield
  - Biological Systems
  - EURACE – EU Economic Modelling

- **CCPForge**
  - Collaborative development environment for CCPs
  - SourceForge-like service based on GForge
  - Funded by JISC to set up initial service
Software Engineering Support Programme (SESP)

SESP activity to provide and encourage the use of up-to-date software engineering techniques and tools within computational science and engineering. The main goals of SESP are:

- accelerate the introduction and widespread use of high-payoff software engineering practices and technology by identifying, evaluating, and maturing promising or underused technology and practices;
- maintain a long-term competency in software engineering and technology transition;
- enable the UK academic community to make measured improvements in their software engineering practices by working with them directly;
- encourage the adoption and sustained use of standards of excellence for software engineering practice;
- foster collaborations with other groups, in the UK, Europe and the US, that have an interest in the applications of advanced software engineering techniques in computational science.
Elements of SESP

- Software Quality Assurance
- Processes for Legacy Software
- Technology Watch
- Evaluation of Methodologies, Tools and Technology
  - Integrated Design Environments
  - Parallelisation & vectorisation software
  - Symbolic Algebra Systems
  - Problem Solving Environments (PSE)
  - GUIs and user interfaces
  - Component technologies
  - ...
Software Quality Assurance

- **Software Quality Assurance** is the basis of software engineering processes that should be undertaken by all software developers.
- **The software life cycle should include:**
  - Requirements gathering
  - Design - software and testing
  - Implementation
  - Testing
  - Deployment
- **The initial target language for most applications is now Fortran 95 or even Fortran 2003.** Although the commercial world of Software QA is dominated by C, C++ and Java, there are good Fortran tools available.
  - PlusFORT, ForCheck and the NAG Ware are but three examples of QA tools for use in implementation and testing.
  - Clearly CVS is the current tool of choice for version control - but there are others. *(Using gCVS and WinCVS can ease the pain)*
Processes for Legacy Software

- For many applications within the science and engineering community the root language has been Fortran 77 and for some - even Fortran 66.
- Software engineering has developed and languages have grown and now Fortran 95 and C provide the main modern vehicles for these applications.
- To maintain and continue to develop the science encapsulated in these legacy codes a process of transformation and re-engineering must be formalised.
- This can be broken into three basic steps:
  - standardisation,
  - transformation and
  - re-engineering.
- SESP has developed a process and gathered a set of tools to aid this transformation process.
A Step-by-step process for legacy software

Legacy Software

Standard-Base Compilation
Transform software into standard compliance

Undesirable Features
COMMON Blocks
Implicit typing
#define/include

Components & OO
Abstraction
Integration

Add New Capabilities
Dynamic memory
Interoperability
Array Operations

Create Interfaces
Wrappers for legacy code
Interfaces for all routines
Evaluation of Methodologies, Tools and Technology

- The computer science community has a long history of developing new methodologies, tools and technologies to aid the development of computing applications.
- These range from new languages, such as C# or JAVA, to frameworks and environments that gather these tools and processes together in an integrated form, such as the Microsoft Visual studio or CodeForge from the Unix world.
- There is a growth in the use of other languages and programming models other than the procedural style of Fortran. Languages such as C++ and Object Orientation are becoming more common in numerical software.
Dissemination of Software and Results

- **All the results of the activity will be disseminated through a CSE Software Engineering Support Programme Web site and through seminars and workshops.** The SESP web site contains:
  - An overview of the aims and objectives of SESP
  - Detail of contacts in the programme
  - Summary pages on the programmes activities and findings
  - All technology watch and assessment reports (pdf, ps, html)
  - Selected software
  - Links to software and other software engineering pages of interest to computational scientists.

- **Seminars and workshops have been arranged to disseminate the results of the activity and to provide hands on experience with specific software tools.**
The SESP web site provides access to:

- Information on software tools
- Documentation on the SESP tool set
- Reports & publication on software engineering
- Links to public domain tools that may be of use
Software tools – acquired or licensed

- ftncheck *(netlib)*
- FORCHECK *(Leiden University)*
- NAGWare Tools *(Numerical Algorithms Group Ltd)*
- plusFORT *(Polyhedron Software Ltd)*
- Understand for Fortran *(Scientific Toolworks Inc.)*
- DDT *(Alinea)*
- VTune *(Intel)*
- A variety of other public domain tools
FORCHECK - Leiden University

Forcheck is the oldest and most comprehensive Fortran verifier on the market.

- It performs a static analysis of Fortran programs or separate Fortran program units.
- Generally Forcheck detects more anomalies in your program than most compilers do.
- Forcheck is ideally suited to get a fast insight in existing and legacy programs.
- It composes documentation of your project with cross-reference tables of each program unit, the complete program and produces a call-tree.
- Forcheck can be used as a software engineering tool in the various stages of the development process and can verify the conformance to each level of the Fortran standard.
- Beside the full Fortran syntax Forcheck supports many language extensions of all popular compilers by compiler emulation.
plusFORT - Polyhedron Software Ltd

plusFORT is a suite of tools for Fortran programmers. The main components are summarized below:

- **SPAG** - The primary analysis and restructuring tool of plusFORT. SPAG processes Fortran 77 with all common extensions, and almost all Fortran 90/95. It can also translate Fortran 77 to Fortran 95
- **GXCHK** - A global static analysis tool
- **CVRANAL** - A coverage analysis reporting tool
- **QMERGE** - A version selection tool
- **QSPLIT** - A small file-splitting utility
- **AUTOMAKE** - A tool for minimal recompilation of Fortran (66, 77, 90 and 95) and C programs.
NAGWare Tools - Numerical Algorithms Group Ltd

- The NAGWare Fortran Tools provide users with the ability to analyse and transform Fortran 77, Fortran 90 and Fortran 95 codes.
- These tools can be used in a range of ways:
  - Quality Assurance - enforcing coding standards
  - Porting to new platforms
  - Converting from fixed format Fortran 77 to free format Fortran 95
  - Normal day-to-day development
- The NAGWare Fortran Tools suite consists of the following components:
  - NAGWare Fortran 95 Tools
  - NAGWare Fortran 77 Tools
Understand for FORTRAN - Scientific Toolworks Inc.

- Understand for FORTRAN is an interactive development environment (IDE) tool providing reverse engineering, automatic documentation, metrics and cross referencing of FORTRAN source code.
- It supports FORTRAN 77 (F77) and FORTRAN 90 (F9X) language standards, with common VAX, Cray and Salford extensions.
- Understand for FORTRAN helps you reverse engineer, understand and maintain large amounts of legacy FORTRAN source code.
- It also includes numerous graphical reverse engineering views designed to help you understand and assess changes you are considering in your code.
The QA Portal - [www.qaportal.cse.clrc.ac.uk](http://www.qaportal.cse.clrc.ac.uk)

- The QA Portal provides a simple web interface to a number of the common software engineering processes
  - analysis
  - transformation.
- It is built on a standard client/server model with user registration and password protection.
- Will process Fortran 77 and Fortran 90/95 software.
- Single files or archive (tar, zip) collections can be processed.
- The results can be viewed or saved to a local file.
- Documentation on the QA Portal and the associated tools is provided online.
Summary

- SESP will provide CSE with software engineering tools and expertise which, to some extent, will be driven by user needs.
- A process for migrating legacy Fortran software has been defined and some software tools identified.
- The legacy process tested on a number of applications with reasonable results - *much more automation is required.*
- A software tools resource has been started - *needs expanding.*
- A web interface to some of the tools has been written - *this need to be expanded and more flexible.*
- There are software tools to aid migration of codes - conformance to a standard source form is the penalty (*not really a penalty*).
- There are tools to help in the understanding and documentation of software - a short learning curve is required.