UT-BATTELLE

SciDAC Tools for Nuclear Energy

Valmor de Almeida
NSTD – Separations and Materials Research Group

31 May 2006

NSTD Summer Seminar

Outline

- DOE SC SciDAC 1 & 2 programs
- Scientific computing tools
  - Common components
- Additional information
  - Geometry and Meshing
  - Solvers

Scientific Discovery through Advanced Computing

- SciDAC-1: 2001 – 2005
  - "Develop new tools and techniques for scientific research through computational modeling and simulation in mission areas of the DOE's Office of Science"
  - Partnership: ASCR and other offices in SC
  - Multidisciplinary & collaborative aimed at <u>terascale</u> computing
    - ✓ Scientific computing infrastructure (math and CS ISIC's)
    - ✓ Science application areas

      Climate, Fusion, High-energy & Nuclear Physics, Chemistry
  - Labs and Universities participants
  - ~$57 M; 51 projects
  - Accomplishments starting to be well documented…….

- **Website: www.scidac.org (PI meetings and Conferences)**
- **SciDAC Review Magazine (first issue)**

SciDAC-2: 2006 – 2010

- Call for proposals closed (Mar '06)
- Greater ORNL participation (NLCF)
- Multidisciplinary & collaborative aimed at petascale computing
  - ✓ Scientific computing infrastructure (CET)
  - ✓ Application areas

    Accelerators, Astrophysics, Climate, Biology, Fusion, Groundwater, …, Radiation Transport, Turbulence,…
- Labs and Universities participants
- ~$67M; 230 proposals (110 universities, 120 labs)
- Announcement of winners late June '06

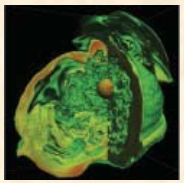
SciDAC-1 Software Development Structure

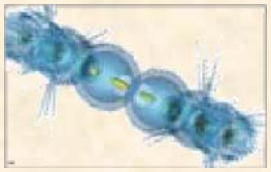
Science Applications ↔ **Integrated Software Infrastructure Centers (ISICs)**

BER | **BES** | **Fusion** | **HEP/NP**

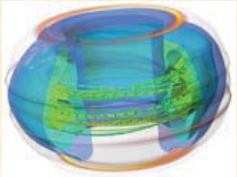
- Climate

- MHD
- Plasmas

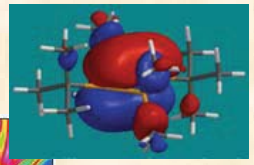


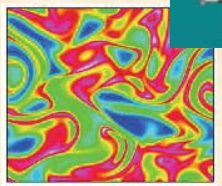


- Astrophysics
- Accelerators



- Quantum chemistry
- Combustion





ASCR

Math

- TSTT
- TOPS
- APDEC

CS

- CCTTSS
- PERC
- SDM
- orthers …

➢ develop scientific simulation software

➢ provide core software

No centralized software repository yet

www.scidac.org

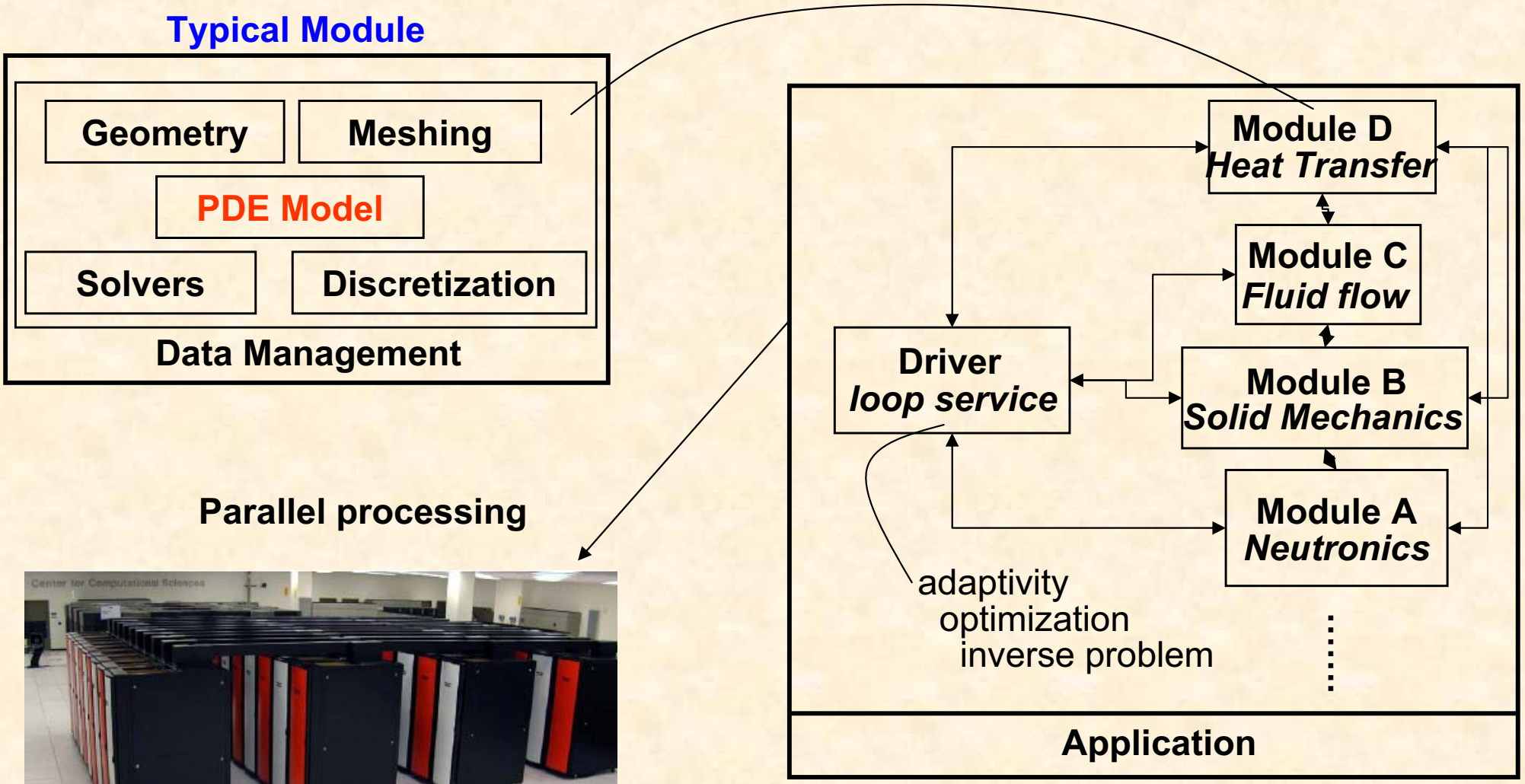
ISICs of Interest to NE Applications

- **CCTTS – Center for Component Technology and Terascale Software**
  - ✓ **Software engineering and integration**
    www.cca-forum.org
- **TSTT – Terascale Simulation Tools and Technologies**
  - ✓ **Geometry, meshing and adaptivity tools**
    www.tstt-scidac.org
- **TOPS – Terascale Optimal PDE Simulation**
  - ✓ **Solvers**
    www-unix.mcs.anl.gov/scidac-tops
    www.scidacreview.org/0601/html/tops.html

Why?

- Tools were developed for scientific applications that share a lot in common to NE applications
  - Algorithm framework for PDE's models (time-dependent, multi-dimensional, multi-physics non-linear, coupled)
  - Design and optimization of mathematical models
- Tools are generic and extensible
  - Likely to be directly applicable to NE applications now or extended to apply to special cases
- ISICs want to continue development in SciDAC-2
  - Tools have 5+ years of development and have a good chance of winning DOE support for another 5 years

Desired Elements of NE Application Software on Terascale Computers



Parallel processing

CCA Common Component Architecture

The Center for Component Technology for Terascale Simulation Software

Main :: Scidac :: { Collaborations }

Quick Links

- CCTTSS Home
- Overview
- Research Agenda
- Participants
- Collaborations
- Sponsor
- Presentations
- Reports
- Outreach
- Contact
- CCA Home

Search

Welcome to the CCTTSS Home Page

The Center for Component Technology for Terascale Simulation Software (CCTTSS) is dedicated to the development of a component-based software development model suitable for the needs of high-performance scientific simulation, particularly the Common Component Architecture (CCA).

The Center is funded by the U. S. Dept. of Energy (DOE) as an Integrated Software Infrastructure Center (ISIC) under the Scientific Discovery through Advanced Computing (SciDAC) program and includes members from Argonne, Livermore, Los Alamos, Oak Ridge, Pacific Northwest, and Sandia National Laboratories, Indiana University and the University of Utah.

This web space includes materials that pertain specifically to the Center, its operations and its activities. Materials about the Common Component Architecture more generally can be found in the CCA Forum web space.

- **Component technology can speed up the process of software development for NE systems**

How Can Common Component Architecture Help?

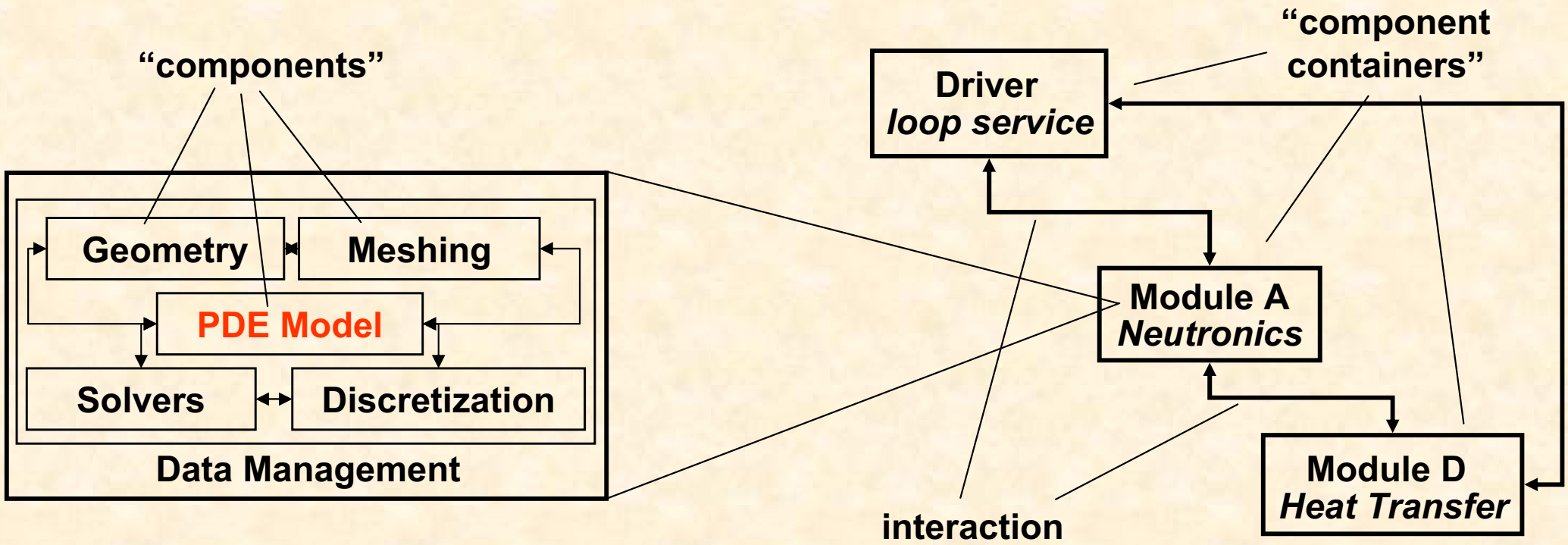
- **Managing code complexity**
  - Code is large and complex; want to make it modular
- **Team work code development**
  - Developers with diverse background working remotely on large, complex codes
- **Coupling codes**
  - Application uses many codes not intended to work together; codes need to remain stand-alone
- **Language interoperability**
  - Codes either mix programming languages or multiple codes are written in different languages

CCA Concepts

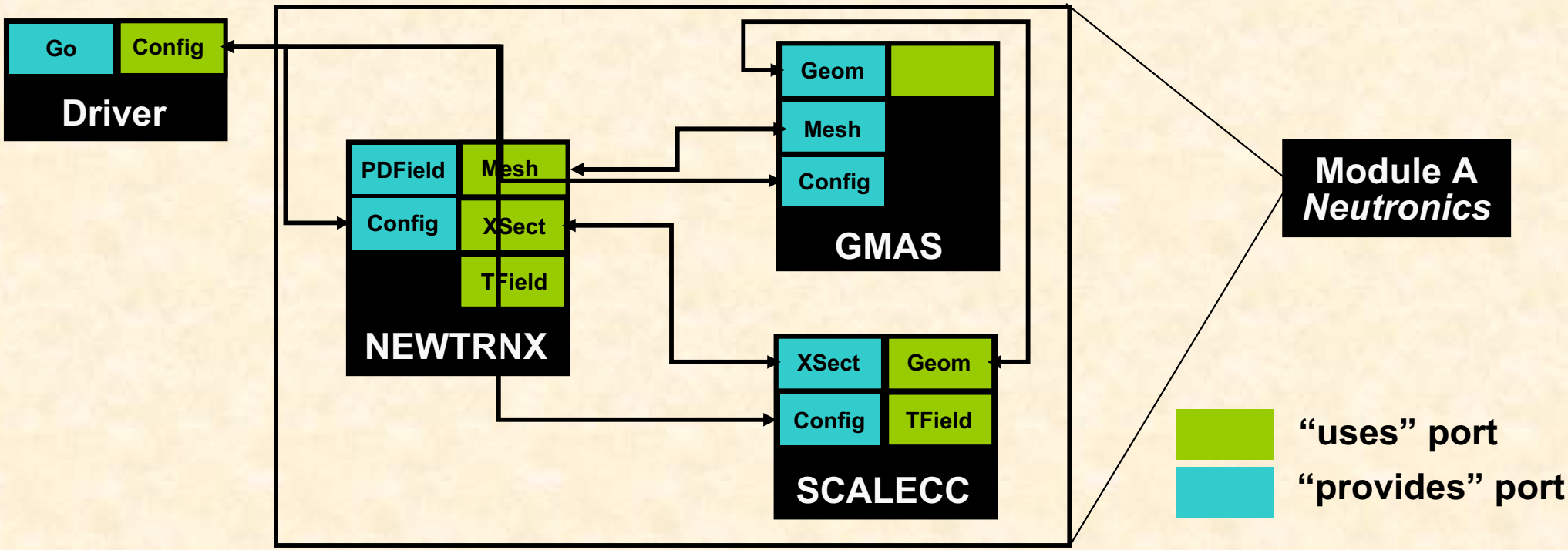
1. Components
   - Interfaces (or ports)
     - Provides port-type
     - Uses port-type
2. Environment
3. Language interoperability

1. What are Components?

- No strict definition but key features:
  - A self-contained software unit with intrinsic functionality
  - Interacts with the outside world only through well defined interfaces
  - Can be composed with (or contain) other components

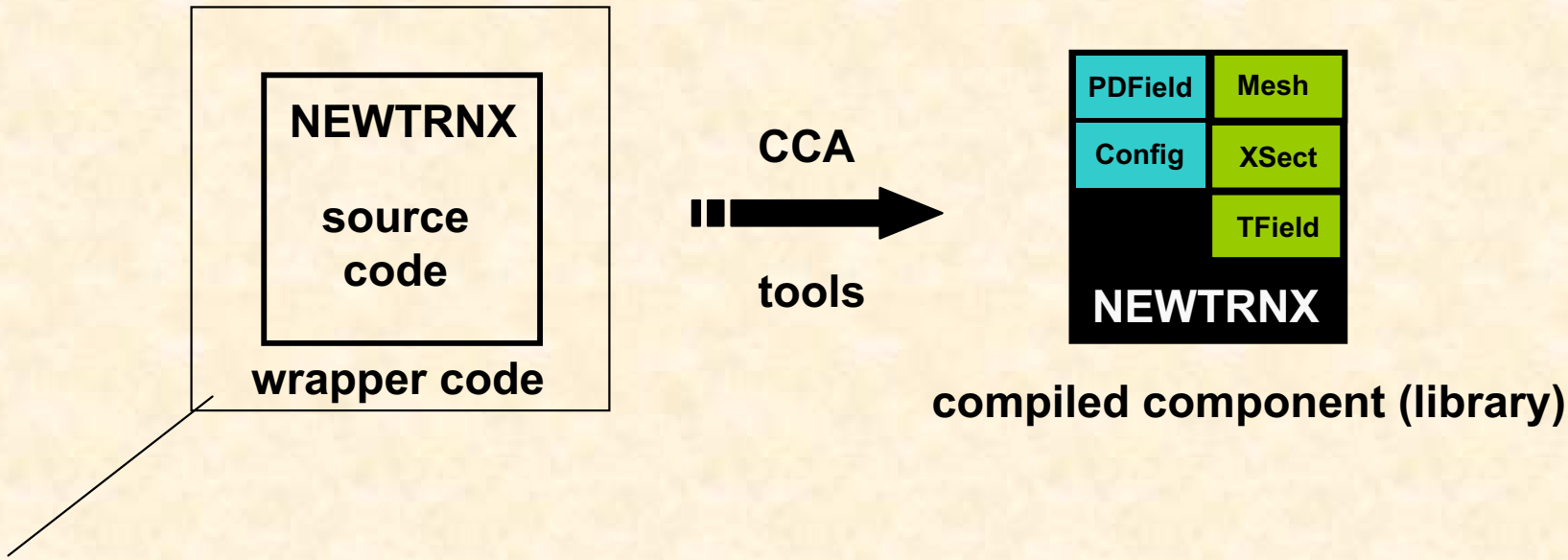


Component internals



- Components interact via well-defined interfaces called "ports"
  - They express computational functionality
  - The "provides" port-type is a FORTRAN module or an OOL class
  - The "uses" port-type is like a subroutine or a function call
  - Data can flow in any direction
  - Ports are designed and programmed by users with a "*neutral-language*" SIDL (Scientific Interface Definition Language; similar to Java).

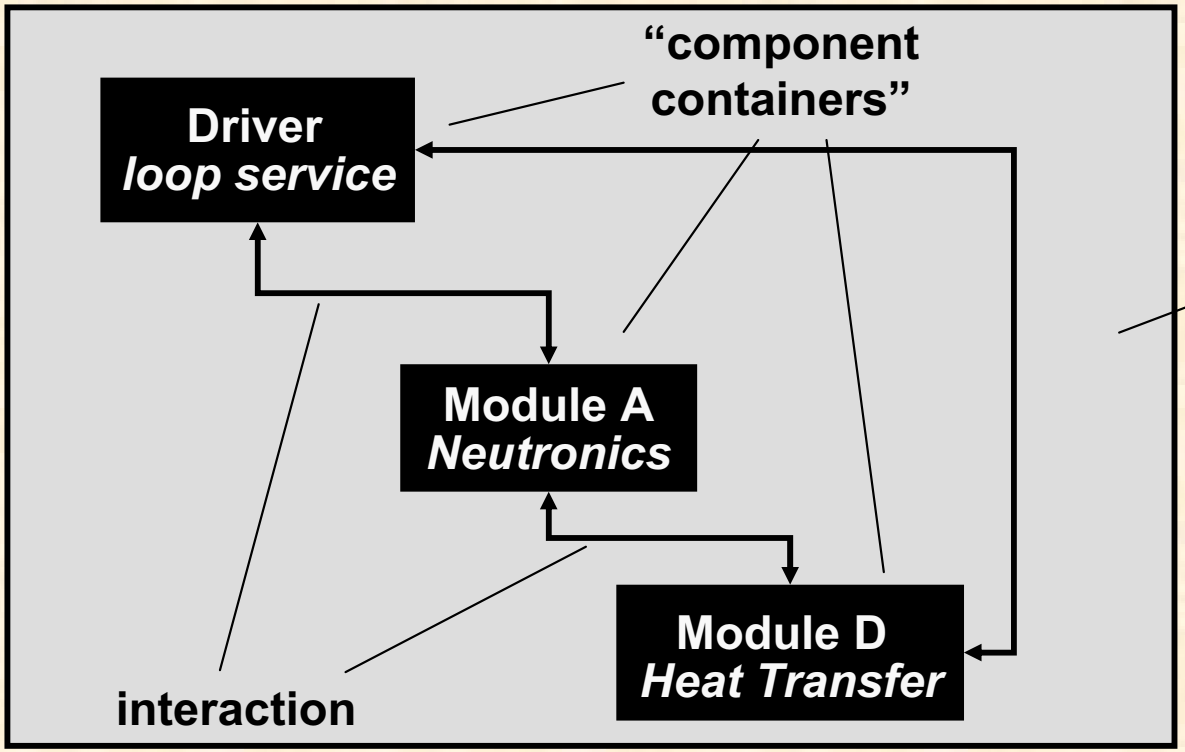
➢ Source code + interface wrapper



- Designed by the user (component and ports definition) in SIDL
- Partially implemented by the user in the same language as the component source code

2. Environment

- The means to “hold” components and “wire” them into applications
- Allow the connection of component ports
- Provide small set of standard services (ports)

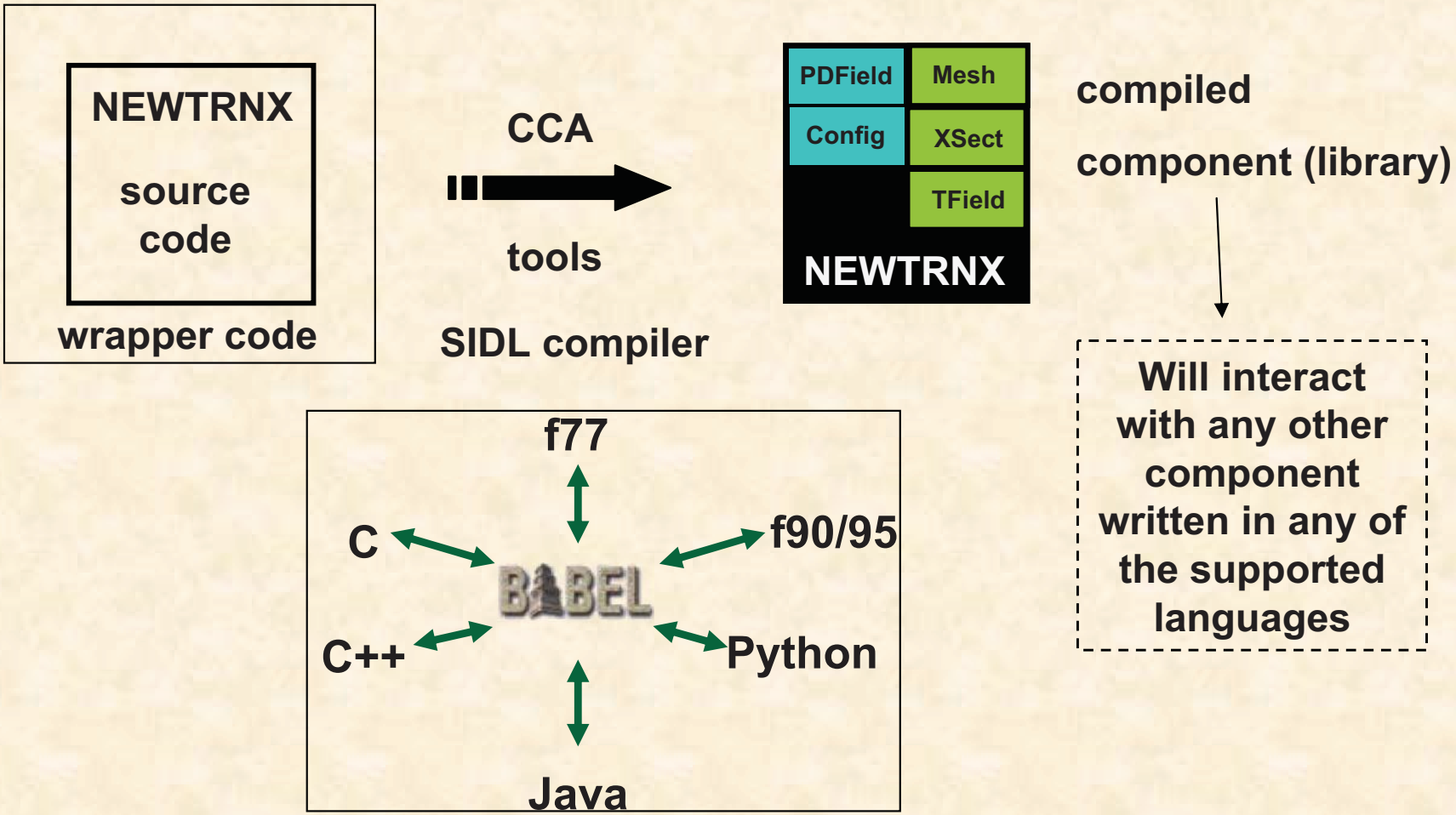


environment

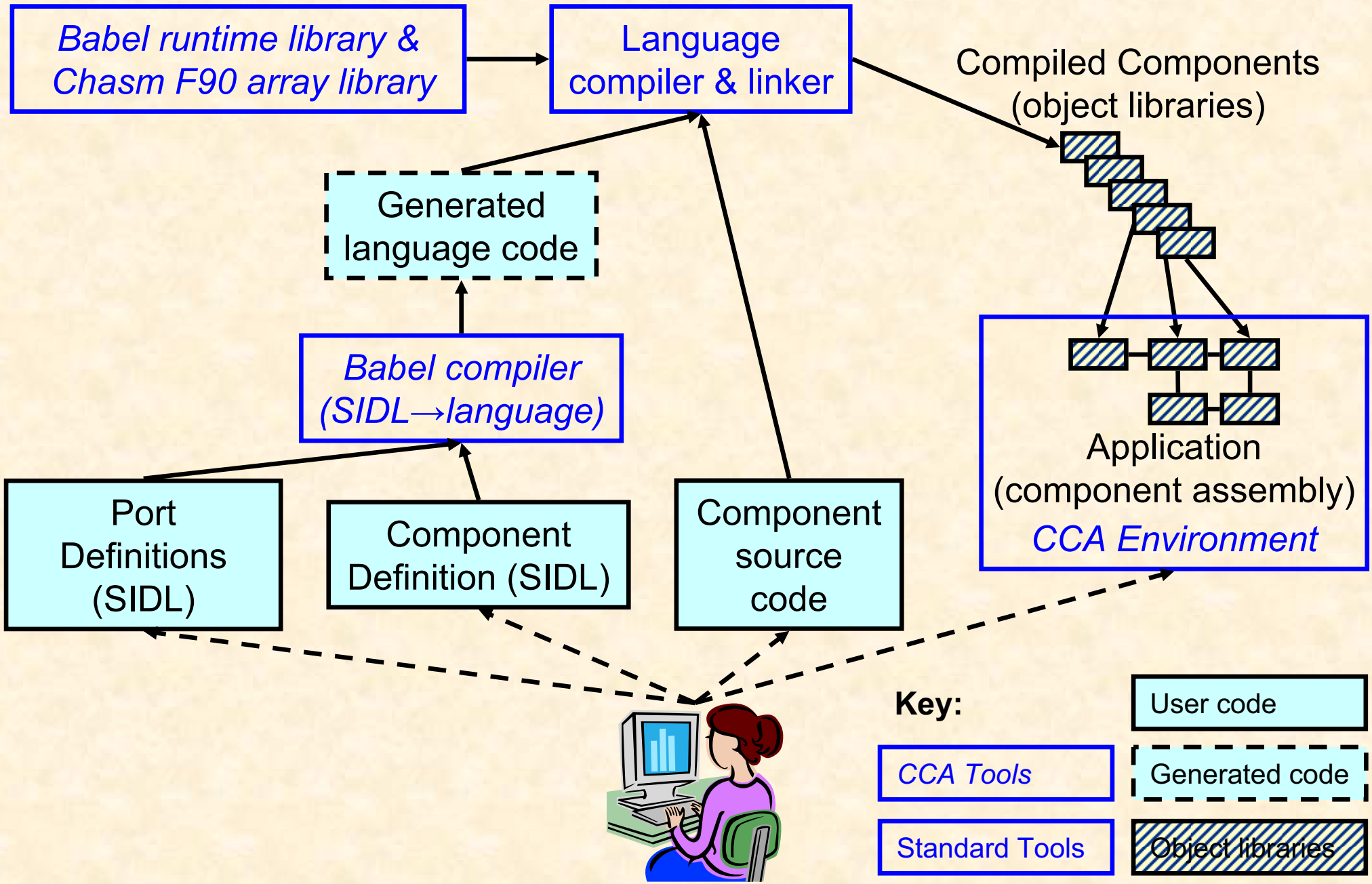
like an elaborated “main” program

3. Language Interoperability

- Environment supports a set of programming languages for components source codes
- A common compiler, Babel, provides the solution to supported languages

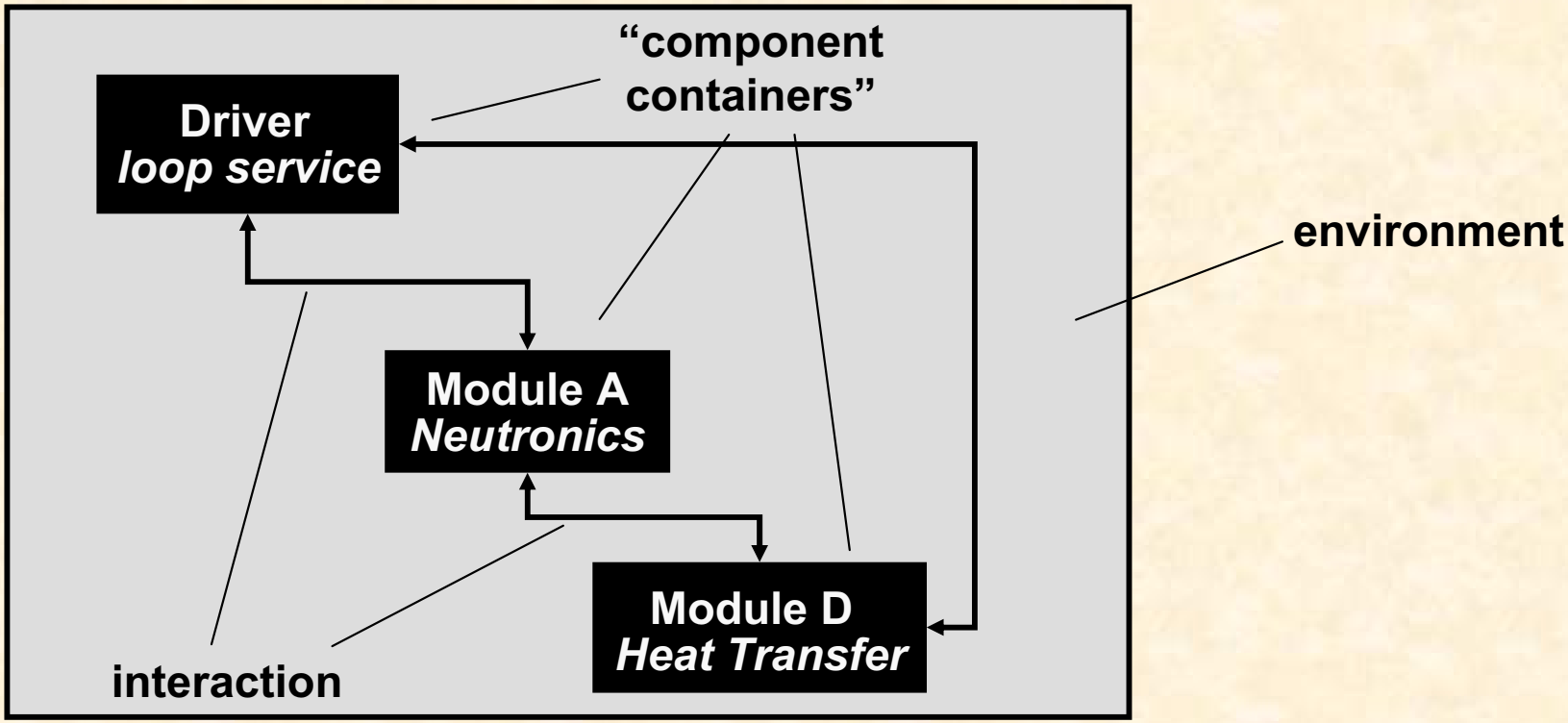


Coding in a CCA Environment



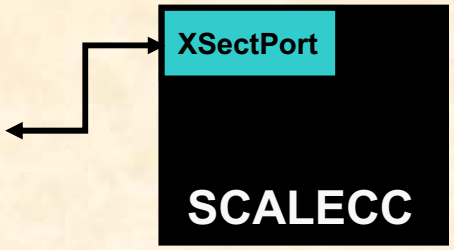
In summary: what is a common component architecture?

- A set of standards that define the:
  - Programming of compatible components
  - Rights and responsibilities of components
  - Expression of component interfaces
  - Environment under which components can be “wired”
  - Rights and responsibilities of the environment



Simple Componentization Example

- Suppose SCALECC is a large code that wants to share an integer type with other components



1. Prototype the port in SIDL

xsection.sidl

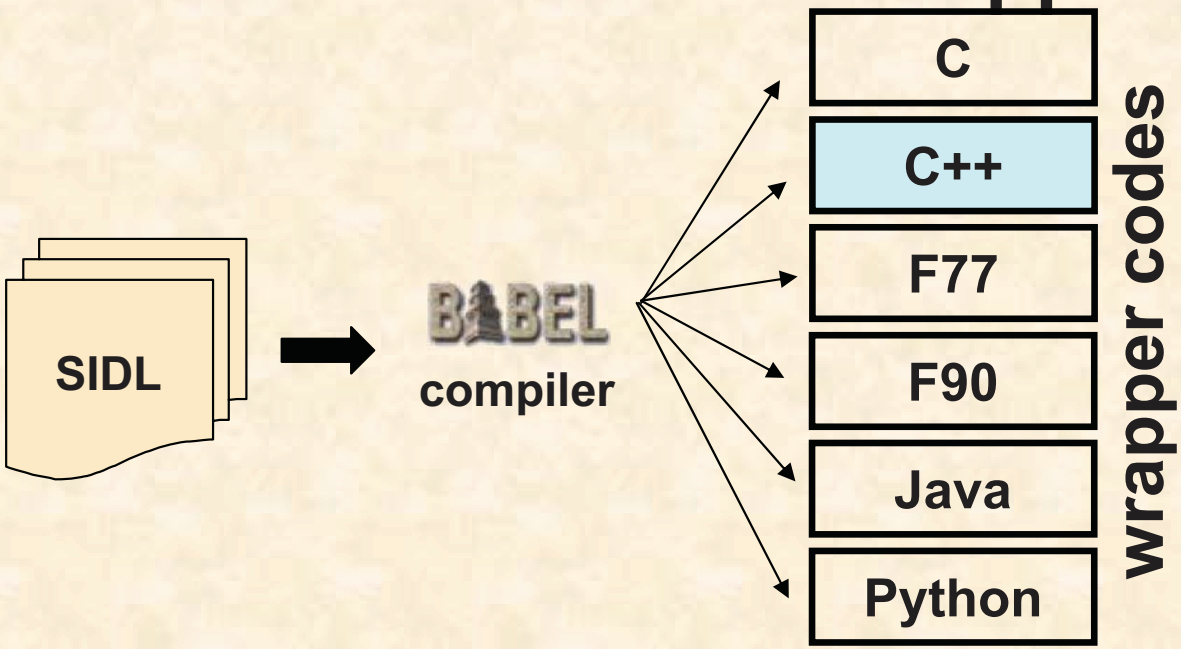
```
package xsect version 1.0 {
    interface XSectPort extends gov.cca.Port {
        int GetNEnergyGroups( );
    }
}
```

2. Prototype the component in SIDL

scalecc.sidl

```
package HTTR version 1.0 {
    class SCALECC implements-all xsect.XSectPort
gov.cca.Component {}
}
```

3. Select the language SCALECC is written on (say C++) and run the SIDL compiler, Babel, on the .sidl files to obtain wrapper codes



4. User finishes the implementation of the C++ wrapper code

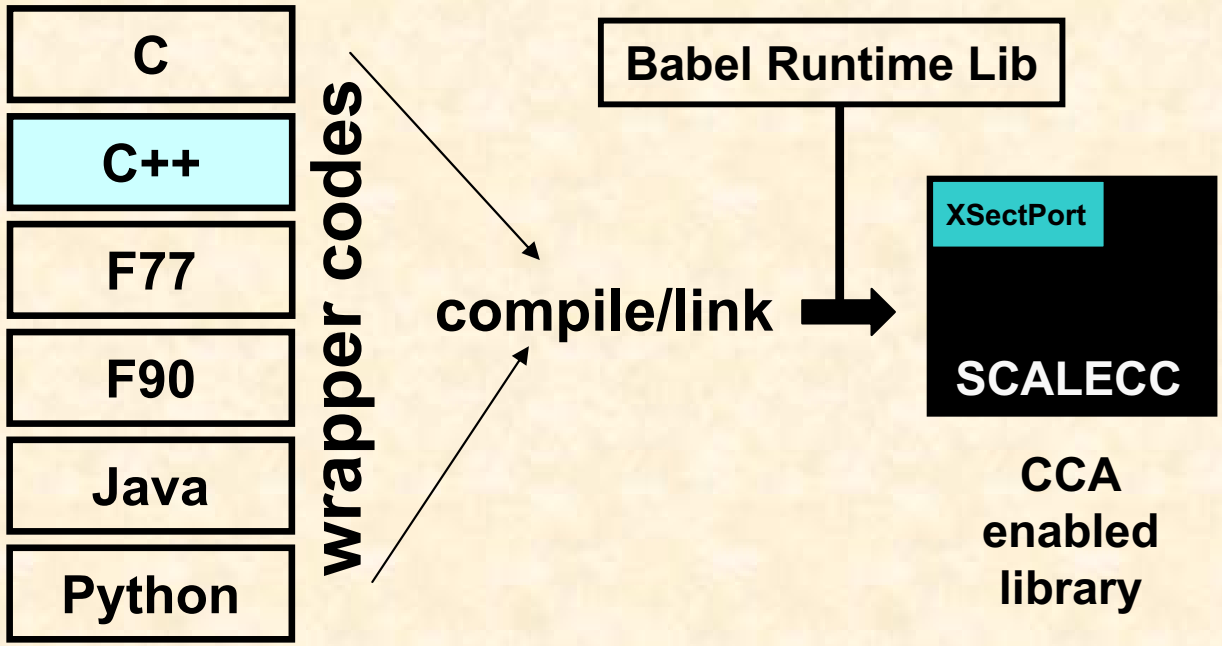
HTTR_SCALECC_Impl.hh

```
namespace HTTR {
  class SCALECC_impl {
 public:
    int32_t
    GetNEnergyGroups() throw ();
  }
}
```

HTTR_SCALECC_Impl.cc

```
int32_t
HTTR::SCALECC_impl::GetNEnergyGroups () throw ()
{
  // DO-NOT-DELETE splicer.begin(HTTR.SCALECC.GetNEnergyGroups)
  // Insert-Code-Here {HTTR.SCALECC.GetNEnergyGroups}
(GetNEnergyGroups method)
  return 123456789;
  // DO-NOT-DELETE splicer.end(HTTR.SCALECC.GetNEnergyGroups)
}
```

5. Compile and link with Babel library



6. Load component into a CCA environment

What is "the" CCA?

- A component environment designed to meet the needs of high-performance scientific computing
  - Support for legacy software
  - Performance is important
  - Parallel computing
  - Support for languages
  - Open source
- Put forward by The Common Component Architecture Forum www.cca-forum.org

CCA Tools Bundle

- cca-tools-0.6.0-rc2
  - babel-0.10.12
  - boost-1.31.0
  - cca-spec-babel-0.7.10
  - ccaffeine-0.5.10
  - ccafe-gui-0.5.6
  - chasm-1.3.rc1
  - makefiles, tutorial, and skeleton directories
- Download:
  - http://www.cca-forum.org/download/cca-tools/

The Common Component Architecture Forum

Software

This page provides links to software pertaining to CCA Forum activites. Links to components can be found in the Component Directory.

CCA Tools

A number of CCA tools are available in single, easy to build tar file (sometimes referred to as the *Sumo Tarball*). This includes everything you should need to build and run your own CCA components:

- **Chasm** - an F90 interoperability library from Los Alamos.
- **Babel/SIDL** - an object oriented language interoperabilty interface definition language.
- **CCA Specification** - The Common Component Architecture Specification for high performance components.
- **Ccaffeine** - a CCA framework compliant with the CCA specification.
- **Ccaffeine GUI** - A Graphical User Interface that works with Ccaffeine.

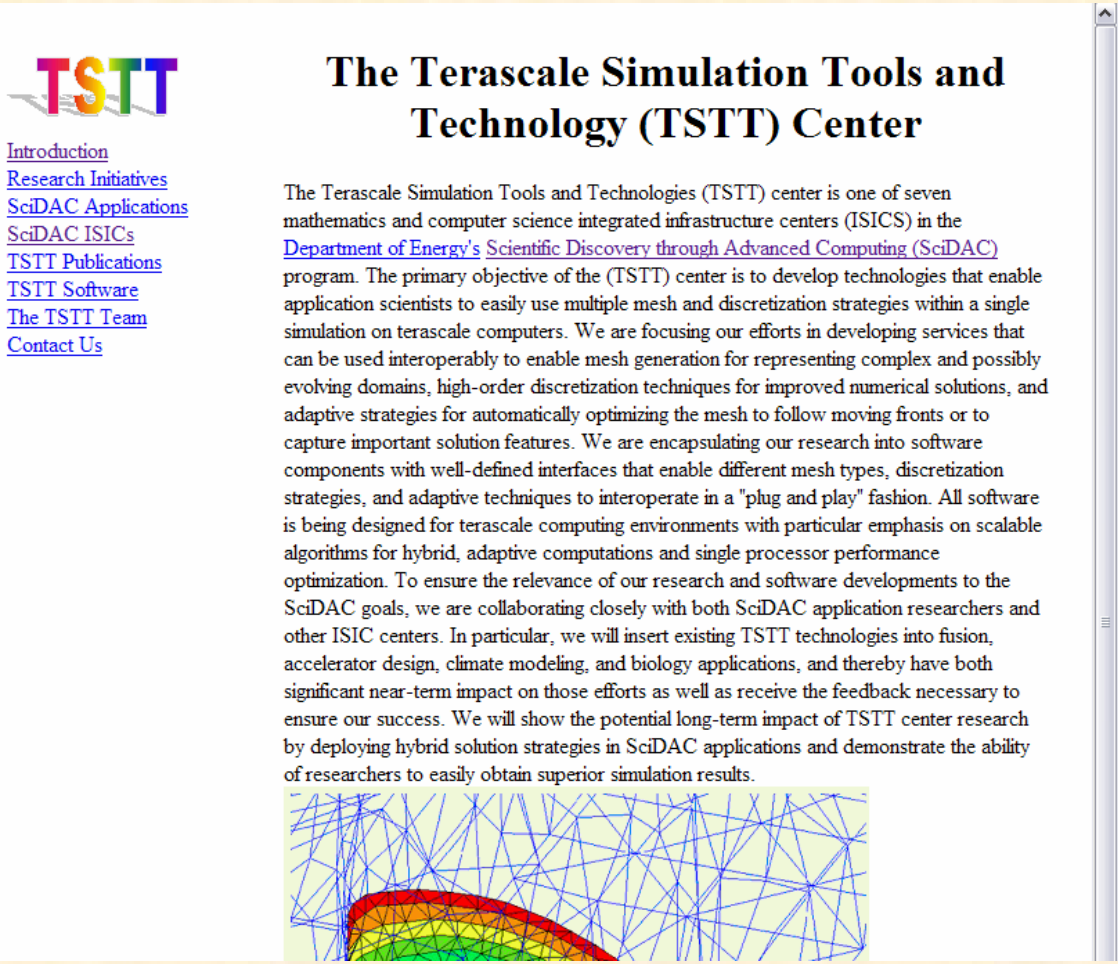
This Sumo tarball also comes with its own build system. *Please read the README that comes with it!*

If your object is to create your own components then this is a great place to

Terascale Simulation Tools and Technology Center

- Software
  - Geometry, mesh relations access
    - CGM, MOAB, LASSO
  - Meshing
    - GRUMMP
    - MESQUITE
  - CCA Interfaces
    - TSTTB
    - TSTTG
    - TSTTM

www.tstt-scidac.org



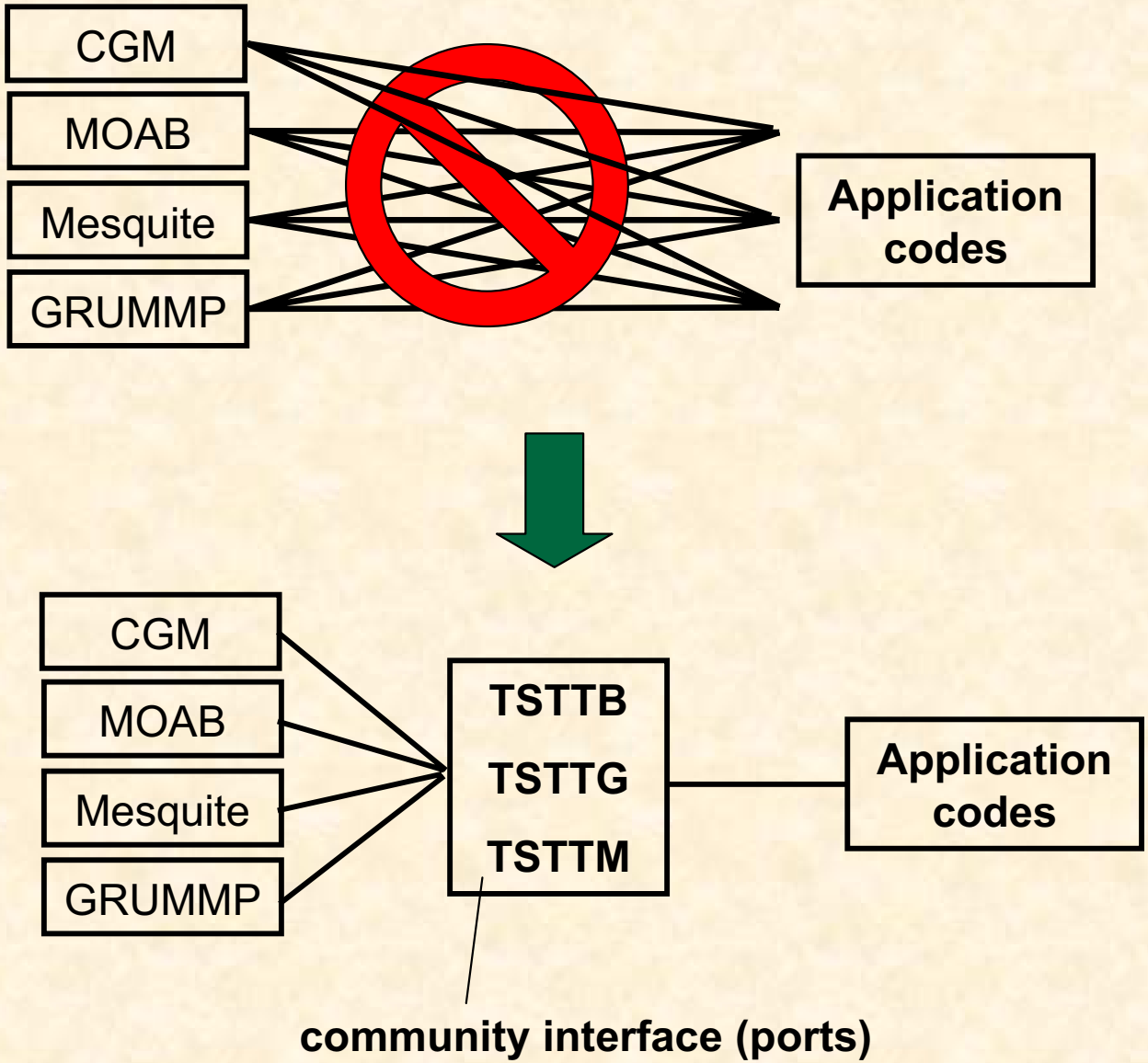
TSTT

Introduction
Research Initiatives
SciDAC Applications
SciDAC ISICs
TSTT Publications
TSTT Software
The TSTT Team
Contact Us

The Terascale Simulation Tools and Technology (TSTT) Center

The Terascale Simulation Tools and Technologies (TSTT) center is one of seven mathematics and computer science integrated infrastructure centers (ISICS) in the Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) program. The primary objective of the (TSTT) center is to develop technologies that enable application scientists to easily use multiple mesh and discretization strategies within a single simulation on terascale computers. We are focusing our efforts in developing services that can be used interoperably to enable mesh generation for representing complex and possibly evolving domains, high-order discretization techniques for improved numerical solutions, and adaptive strategies for automatically optimizing the mesh to follow moving fronts or to capture important solution features. We are encapsulating our research into software components with well-defined interfaces that enable different mesh types, discretization strategies, and adaptive techniques to interoperate in a "plug and play" fashion. All software is being designed for terascale computing environments with particular emphasis on scalable algorithms for hybrid, adaptive computations and single processor performance optimization. To ensure the relevance of our research and software developments to the SciDAC goals, we are collaborating closely with both SciDAC application researchers and other ISIC centers. In particular, we will insert existing TSTT technologies into fusion, accelerator design, climate modeling, and biology applications, and thereby have both significant near-term impact on those efforts as well as receive the feedback necessary to ensure our success. We will show the potential long-term impact of TSTT center research by deploying hybrid solution strategies in SciDAC applications and demonstrate the ability of researchers to easily obtain superior simulation results.

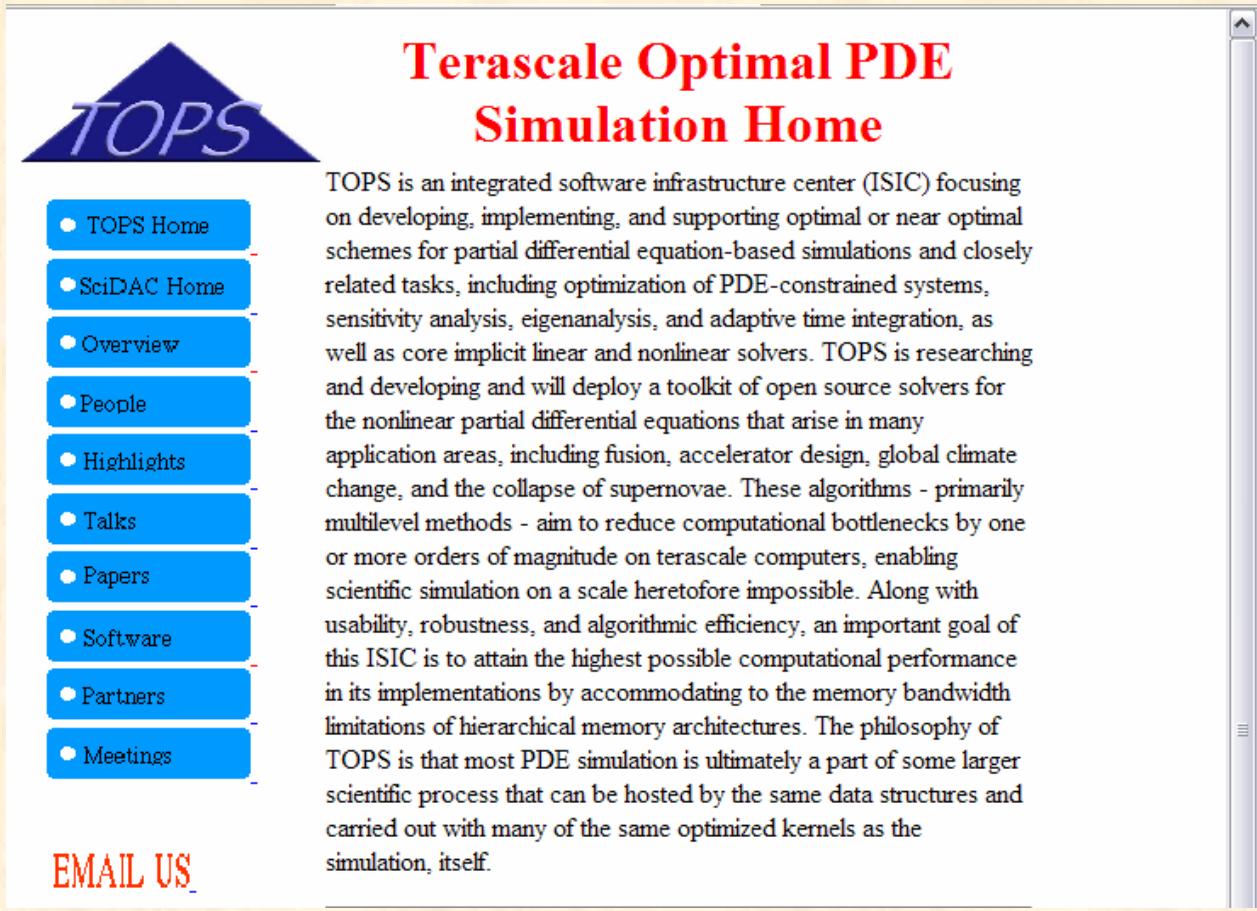
➢ Interfaces are Key to Reuse and Interoperability



Terascale Optimal PDE Simulation Center

www-unix.mcs.anl.gov/scidac-tops

- Software
  - Preconditioners
    - Hypre
  - Eigensolvers
    - PARPACK
    - SLEPc
  - Non-linear/ linear solvers
    - PETSc
    - SuperLU
  - ODE integrators
    - SUNDIALS
  - Optimization
    - TAO
    - Veltisto
  - CCA Interfaces
    - Not available yet



QUESTIONS?

Slides at:

http://neutronics.ornl.gov/scidac_ne.pdf

➢ TOP500 List of Supercomputers: Nov. 2005

1-100 | 101-200 | 201-300 | 301-400 | 401-500

Rank	Site	Computer	Processors	Year	R_{max}	R_{peak}
1	DOE/NNSA/LLNL United States	BlueGene/L - eServer Blue Gene Solution IBM	131072	2005	280600	367000
10	Oak Ridge National Laboratory United States	Jaguar - Cray XT3, 2.4 GHz Cray Inc.	5200	2005	20527	24960

classified use

open research use

➢ First US open research teraflop computer: Nov. 2000

8	IBM SP Power3 375 MHz 8 way/ 1152	929.00 1728.00	UCSD/San Diego Supercomputer Center USA/2000

➢ First teraflop computer: June 1997

classified use

Rank	Manufacturer Computer/Procs	R_{max} R_{peak}	Installation Site Country/Year
1	Intel ASCI Red/ 7264	1068.00 1453.00	Sandia National Labs USA/1997