Software Integration and Support

J.R. Cary

September 23, 2010

From March:

The PAC endorses a staged software delivery model with early and periodic releases, each with greater capability. This will be important for community support and feedback.

F Brooks (1995):

When I first wrote The Mythical Man-Month in 1975, I counseled programmers to 'throw the first version away,' then build a second one. By the 20th-anniversary edition, I realized that constant incremental iteration is a far sounder approach. You build a quick prototype and get it in front of users to see what they do with it. You will always be surprised.

This is a work in progress. More after framework team incorporates results of science drivers.

Charge

1) FSP Science Goals -- Regarding the current set of science drivers and associated science program plans, please comment on the current vision and plans for:

   a) development of the integrated software products that flow from the science goals in the nearer-term (<5 years) and the longer term time frames (5-15 years);

   b) prioritizing and scheduling the program elements identified in the individual science program plans; and

   c) the cross-disciplinary engagement of communities representing FES theory/computations & experiments and ASCR computer science and applied math.

2) FSP Management & Governance -- Regarding the FSP Execution Plan, please comment on the currently proposed approach for:

   a) organizational structure, along with the defined roles and responsibilities and the decision-making processes

   b) interactions and interfaces in view of the distributed nature of the FSP project and its dependences on external collaborations;

   c) budgeting and associated flow of funds/resources from DoE-SC to the lead institution and to the collaborating research performers at other laboratories, universities, and industries.

FSP Operational Principles

- **The FSP mission is to deliver computational software for addressing physics problems that integrates multiphysics over different regions or scales.**
- **The FSP will lead to savings by maximizing reuse and providing common infrastructure**
- **The FSP will leverage existing fusion community investment**
- **The FSP should follow best practices from other communities (climate being notable) (best = successful)**

Lead with the science, develop Integrated Science Applications

Science Drivers

- Boundary Layer
- Pedestal
- Core Profiles
- Wave-Particle Interactions
- Disruptions
- Whole Device Modeling

Will not repeat coupling discussions of Martin, Xianzhu
Beware: "Code" and "application" are both highly overloaded.
FSP has adopted the terminology *Integrated Science Application*

Will need a range of couplings: dimensionality and implicitness

Integration types	Occurs in	Comments	Rel
0D, implicit	Core-edge integration (pedestal)	First release is for exp. derived edge transport and simple edge model, second relies on validated edge component.	1, 2
Multiple 0D, implicit	Embedded turbulence, edge-wall interaction	Both linear and tree connectivity	1
1D, explicit/implicit	core transport, communicate to eq.	implicit needed for higher-order algorithms	1
2D, explicit/diagnostic	WDM, stability analysis of evolving system		1
2D, implicit	Sources in WDM, needed for > 1^{st} order		1

Rel 1 = ± year 2, Rel 2 = 5-7 years

Will need a range of couplings: dimensionality and implicitness

Integration types	Occurs in	Comments	Rel
3D, tight or implicit	Wave-particle, neutrals in edge turbulence codes, 3D Eq., non axisymmetric walls, WDM with mesoscale dynamics	Within components now, work to common structure This is a BIG area. Early releases based on one-offs	1,2
N-D, diagnostic	Data analysis viz,	Typically file based, but could be stream based Development of I/O standards, metadata standards.	1,2

Rel 1 = ± year 2, Rel 2 = 5-7 years

We need to take into account how software is successfully developed

- Successful projects have their focus on one target (science driver) with their peripheral vision covering many more.
- It is roughly 8 years from initiation to widespread use (Post)
- Launch N Integrated Science Application (in sense of computational application) teams to cover some number of targets. (FYI, CESM ~ $4-5M/yr)
- By what process do we create these teams that maintains any *team fusion* (not D-T) from existing fusion community teams? (DeMarco and Lister, also F. Brooks, "it is the breaking of fusion of the old team that aborts the embryonic product.")

Process for creating an Integrated Science Application team

Science Drivers
Boundary Layer
Pedestal
Core Profiles
Wave -Particle Interactions
Disruptions
Whole Device Modeling

- **Prior to December 31, 2010 (noting that we have two face-to-face meetings), we will have decide upon the scope of work for the first X years and we will have broken this down into N sets of goals and requirements, derived from the Science Drivers and acting as first goal set for an integrated application team.**
- **These will be presented and discussed at a community planning workshop in Jan 2011.**
- **By Feb 2011, IA heads will be selected.**
- **IA heads form groups that are vetted by management.**
- **Each IA head runs group to do an improved estimation.**

Integrated applications and Integration Team

Science Drivers	Software Integration
Boundary Layer	
Pedestal	
Core Profiles	
Wave -Particle Interactions	
Disruptions	
Whole Device Modeling	

From March:

The PAC endorses a staged software delivery model with early and periodic releases, each with greater capability. This will be important for community support and feedback.

Operational Principles

- **The FSP will lead to savings by maximizing reuse and providing common infrastructure**
- **The FSP will leverage existing fusion community capital**
- **The FSP should follow best practices from other communities (climate being notable) (best = successful)**

Conclude: create integration team to provide common infrastructure

Software integration team

analogs: ESMF ($4-5M/yr), software reuse teams

Software integration

- Mission: to provide a common code base for developing integrated applications
  - Enabling software and tools (physics, math, visualization, data analysis)
  - Workflow (scripting, run management, visual programming)
  - Application integration (distributed data structures); Data spaces to be "evaluated" as part of the research program.
- Assessment: Are the supported tools being adopted?

Enabling software, tools, and technology

- Mission: to develop libraries to assist software development and tools to assist users throughout the program
- Examples: physics libraries, math libraries, I/O libraries, uniform I/O interfaces, metadata, viz, data analysis.
- Can and should be highly distributed (e.g., I/O team at one institution or one set of same), and will be largest of the three support teams
- Enabling software and tools stripped from the protoFSPs and elsewhere will be supported and managed here.
- Assessment: Are the supported tools being adopted?

Create workflow team

- **Mission: to assist users in end-to-end simulations: *idea* to *reported result***
- **Examples: script-based tools, visual programming (Kepler, VizTrails), traditional GUI (Visit for workflow)**
- **Could be distributed (e.g., Kepler support at one place) on a per-technology basis.**
- **Workflow tools stripped from the protoFSPs and elsewhere will be supported and managed here.**
- **Assessment: Are the supported tools being adopted by users?**

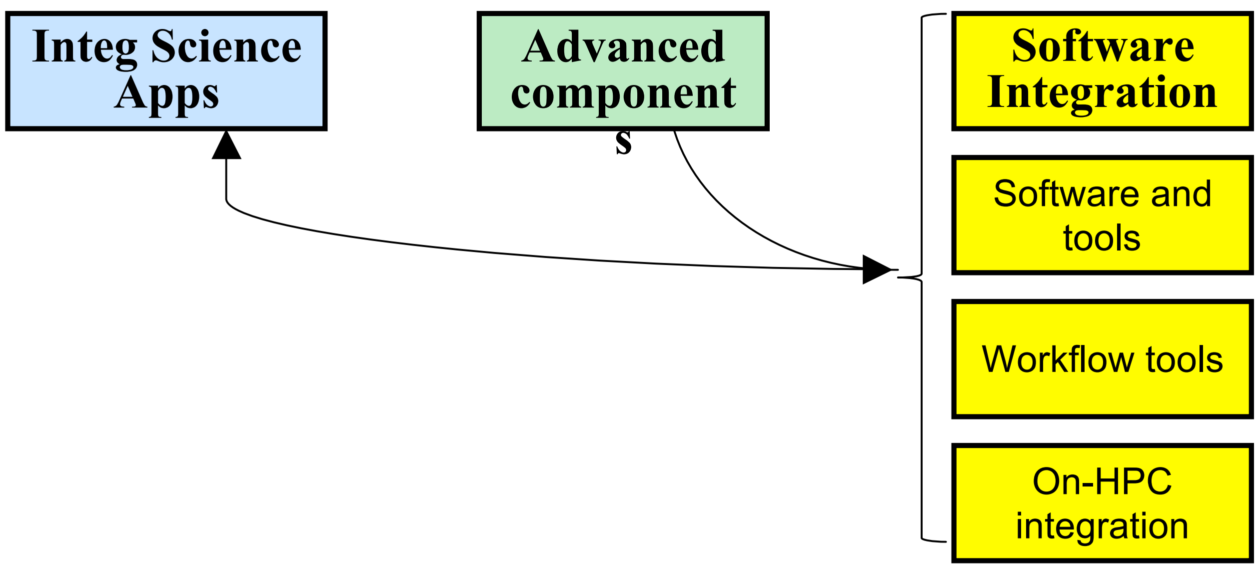
Create on-HPC application integration team

- Mission: to develop the tools usable in the on-HPC environment for tight coupling
- Examples: distributed data structures, intergrid mapping technology
- Should be more collocated
- Should have liaisons with each integrated application. Should extract best practices from all integrated apps and feed those back out.
- Assessment: Are the supported tools being adopted by the integrated applications? (But assess on 5-year time scale)

How will *Software Integration* work with other teams?

- Short term success will be made by modifying existing community software to address science drivers. This is likely to be less invasive on legacy software and more one-offs.
- Software integration liaisons will study the efforts, extract common tools, and make them more broadly applicable and available
- Software Integration will also be identifying and developing tools for tighter couplings, such as common data structures
- All of this will be maintained for public download

Development of robust, integrated apps on long term will require continuous interaction



Integration ideas flow from integrated apps to the Integration team
Integration team creates robust, reusable integration software

Timeline for integration and support teams

2012 | 2014 | 2016 | 2019

Integrated Science Application 1

Integrated Science Application 2

Integrated Science Application 3

First release

Tightly integrated app release

Software integration

First release (ext tools, viz, …)

Second release (tight coupling infrastructure)

Operations

Create multiple permanent *efforts* (how large? reporting?)

- Operations
  - Developer support team
  - User support team
  - System support? (Should we have facilities?)
- Quality management team
  - V&V, UQ methodologies
  - Test systems
  - Validators

Create a developer support team

- **Mission: to develop tools for supporting development, help with toolizing, and administer repositories, bug tracking, performance regression**
- **Examples: customize and adapt build systems, administer svn repos, will need support for authentication and authorization (commong)**
- **Could be highly distributed (e.g., different personnel for each task area). Will not require (many) PhD scientists.**
- **Development supporting tools from the protoFSPs and elsewhere will be supported and managed here.**
- **Assessment: Are the supported tools being adopted?**

Create a user support team

- Mission: to help users use the developed tools, understanding that in some cases this involves triaging.
- Examples: develop documentation for use and deployment, assist users in use at central facilities and in local deployment, ticket system.
- Could be highly distributed (e.g., different personnel for each task area). Will require (mostly) developers and PhD scientists. Start smaller, grow.
- External deployment practices should be studied for potential adoption. Assessing data to improve processes.
- Assessment: User satisfaction

Create a system support team? (if we have facilities)

- Mission: ensure the systems keep running, to assist users with system use issues.
- Examples: develop documentation for login, remote execution, hold workshops.
- Could be a mix of local and remote, but should have a strong local team.
- Assessment: User satisfaction

Create a quality assurance team

- **Mission: to provide software methodologies for software testing in all aspects: regression, verification, validation (pool of validators for the IA teams)**
- **Methodologies team: PhD level scientists who do research in this area and also evaluation, work to improve methods for V&V, UQ**
- **Testing team: develop/adapt regression/unit systems, maintain (accept same) same as software evolves, tests cross deployment software, workflow software.**
- **Validators: either housed here or in app teams, but must have a common home for information exchange, identify weaknesses.**
- **Could be highly distributed (e.g., different personnel for each task area). Will require a mix of developers (test updating) and PhD scientists (V&V, UQ).**
- **Assessment: User satisfaction with the software, papers written by methodology personnel, validators.**

Timeline for all teams

2012 | 2014 | 2016 | 2019

Developer support

User support

System support?

Quality assurance

Mission refine, hiring

Maintenance

Integrated Science Application 1

Integrated Science Application 2

Integrated Science Application 3

First release

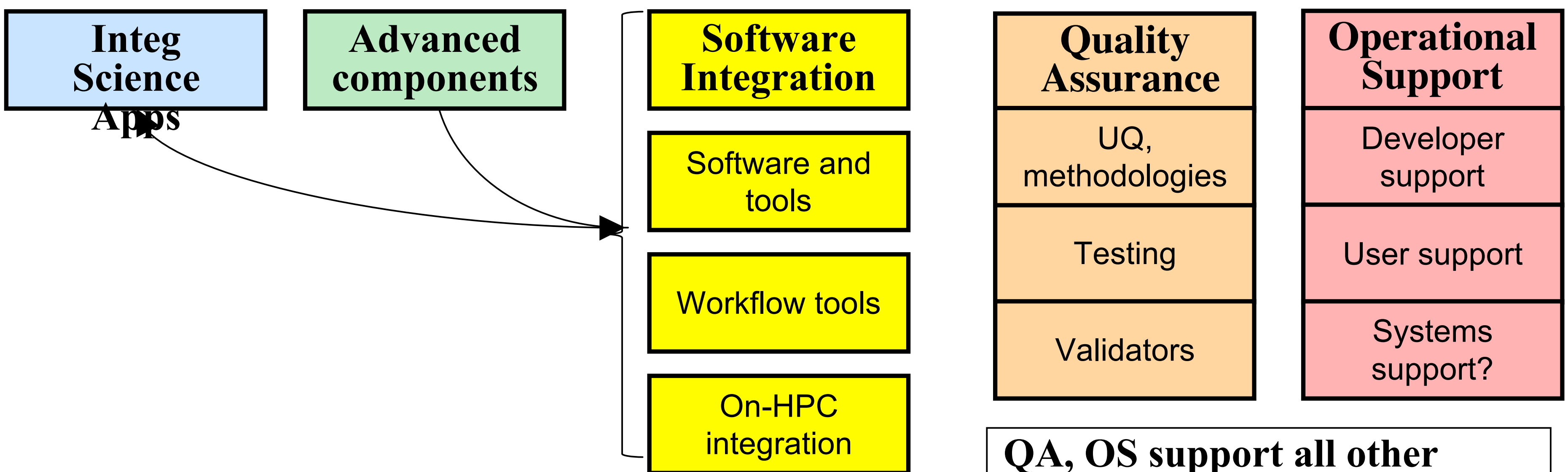
Tightly integrated app release

Software integration

First release (ext tools, viz, …)

Second release (tight coupling infrastructure)

Integrated Science Apps will have assistance from QA and operations



QA, OS support all other areas and users.

Decisions

which boxes have technical managers?

How large are the teams?

Charge

1) FSP Science Goals -- Regarding the current set of science drivers and associated science program plans, please comment on the current vision and plans for:

a) development of the integrated software products that flow from the science goals in the nearer-term (<5 years) and the longer term time frames (5-15 years);

In the first five years, integrated software products will come from stripped, augmented, and reorg'd protoFSPs.

After first release of tight-integration products, new integrated applications will come from refactoring to make use of common infrastructure

b) prioritizing and scheduling the program elements identified in the individual science program plans; and

c) the cross-disciplinary engagement of communities representing FES theory/computations & experiments and ASCR computer science and applied math.

New integration products will make heavy use of AppMath libs and development will be with joint teams involving ASCR personnel.

END

Science Drivers

Boundary Layer

Pedestal

Core Profiles

Wave -Particle Interactions

Disruptions

Whole Device Modeling

Extant robust *Boundary Layer* work has separate fluid edge turbulence codes without neutrals, no wall coupling. FSP could provide

Integration Physics	Relies on	Integration/common needs	Yr
Plasma-neutral coupling (2D)	EM fluid turbulence		1-2
SOL-wall interaction/hydrogenics	Dynamic wall model (H/D charging and discharging)	Point-wise coupling with wall model	1-2
Plasma-neutral coupling (3D turbulence)	3D turbulent transport code with neutrals or coupled 2D transport with 3D turbulence.	Extracted, common, atomic physics libraries, photon transport, 2D or more coupling of fluxes to transport	2-5
SOL-wall interaction, materials, erosion, redeposition	MD model for wall	Point-wise coupling with wall model	5-15

Science Drivers

- Boundary Layer
- Pedestal
- Core Profiles
- Wave -Particle Interactions
- Disruptions
- Whole Device Modeling

Pedestal effort: determine height, size and effect of ELMs, how driven by core, interaction with wall (overlaps with Boundary Layer)

Integration Physics	Relies on	Integration needs	Yr
Coupling core and time-averaged pedestal	Robust simple model of average pedestal height	0D coupling with core evolution	2
Turbulence and transport during quiescent buildup	Validation of existing models; robust free boundary eq. solver accurate through SOL	QL coupling from turbulence code to 2D transport code? Neutrals in EM GK?	1-2
Effect of ELMs on the walls.		Similar to boundary layer	2-5

Science Drivers

Boundary Layer

Pedestal

Core Profiles

Wave -Particle Interactions

Disruptions

Whole Device Modeling

Core profiles brings subscale coupling and mesoscale phenomena

Integration Physics	Relies on	Integration/common needs	Yr
Local transport models	EM-GK turbulence computations.	Embedded subscale, 0D	2
Mesoscale transport	3D magnetic eq., new formalism for incorporating transport into same.	Formalism needed first.	10

Science Drivers

Boundary Layer

Pedestal

Core Profiles

Wave -Particle Interactions

Disruptions

Whole Device Modeling

Extant wave-particle work has separate computational applications for RF propagation codes, particle transport,

Integration Physics	Relies on	Integration/common needs	Yr
Alfvenic instabilities in presence of energetic particles	Computations of alfvenic modes.	3D coupling of eigenmode fields with 1D coupling of amplitudes.	2
Linear RF-plasma coupling at edge	Nonlinear description of RF-edge coupling.		
Nonlinear RF-plasma coupling at edge		3D data exchange in overlap region. NxM.	5

Science Drivers

- Boundary Layer
- Pedestal
- Core Profiles
- Wave -Particle Interactions
- Disruptions
- Whole Device Modeling

Disruptions: Extended MHD development plan

Integration Physics	Relies on	Integration/common needs	Yr
FP for runaway electrons	Robust, flexible equilibria code	Not a true coupling envisioned at present due to formulation issues	2
Structural mechanics codes	Models of wall to varying degrees	Tight coupling of surfacial coupling	2-10
Material walls	Models for wall behavior	Surfacial coupling	2-10
Plasma Control System (PCS)	Ability of PCS to call plasma models	Tight control of algorithms	
Sources and flux models	Source codes and reduced flux models	Improved component interfaces to go to 2D/3D	2-10

Science Drivers

Boundary Layer

Pedestal

Core Profiles

Wave -Particle Interactions

Disruptions

Whole Device Modeling

Disruptions: WDM-based development plan

Integration Physics	Relies on	Integration/common needs	Yr
Evaluation of MHD stability	Robust, flexible equilibria code	Fast evaluation of perturbed equilibria and multiple MHD code runs for each equilibrium	2-10
Runaway electron modeling using F-P codes	F-P component that includes terms important for REs	Typically 1D communication of profiles	2
Material walls	Models for wall behavior	Surfacial coupling	2-10
Plasma Control System (PCS)	Ability of PCS to call plasma models	Tight integration of control algorithms	2
3D equilibria	Robust and validated 3D eq codes	1D profiles to 3D eq	2-10

Science Drivers

- Boundary Layer
- Pedestal
- Core Profiles
- Wave -Particle Interactions
- Disruptions
- Whole Device Modeling

Whole-device modeling brings additional couplings

Integration Physics	Relies on	Integration/common needs	Yr
Turbulence on transport time scales	Gyro-kinetic and integrated modeling codes	Evolution of plasma profiles, including turbulence	3
Interaction of boundary with plasma core	1-1/2-D core and 2-D edge codes	Plasma and neutrals transport, atomic physics	6
3-D free-boundary plasma evolution	3-D equilibrium with magnetic islands and stochastic fields	3-D equilibrium, sources, sinks, transport	9
Prediction, control and mitigation of instabilities	Macroscopic instability codes	Nonlinear macroscopic instability together with integrated modeling	12