Report of the
Proto-FSP Assessment Panel
For the FSP

Final
25 Oct. 2010

Executive Summary

The assessment panel found that each of the Proto-FSPs has interesting and relevant technology that may be useful as *prototypes* of a variety of concepts for the FSP. Some of the Proto-FSP ideas and elements may be the foundation for further development by the FSP. The panel provides a number of detailed recommendations for the FSP in the areas of project management, numerical algorithms, frameworks and software architecture, and physics components and users, based on the experiences of the Proto-FSPs combined with our experiences on other projects. These can be grouped in seven principal recommendations:

- Organize the FSP effort about science drivers, including their reference experiments,
- Produce a validated, robust set of Open Source codes available to the fusion and broader community,
- Form multi-disciplinary teams to address the breadth of science challenges,
- Adopt project management strategies that are suitable for-the scale of the FSP,
- Pursue a single software framework,
- Establish documented software development and quality assurance practices that reflect the investment being made by the FSP and the value of the decisions it seeks to influence, and
- Expect that the design and implementation will require multiple iterations.

The assessment panel does not intend that its recommendations for the FSP become a burden on the Proto-FSPs. At the same time, the panel strongly recommends that the FSP *not* be expected to continue support all of the technology developed under the Proto-FSPs.

The science drivers and their reference experiments for the FSP must set the requirements for the physics, computer science, applied mathematics, and software technologies. A successful FSP project must include a broad assessment of technologies available, both in proto-FSPs and in other projects and communities, to address the needs derived from these drivers. The panel recognizes that significant challenges in multi-physics modeling and integration remain and will require novel spatial and temporal coupling techniques. Appropriate research and development are essential aspects of the FSP. The assessment panel recommends that the management and review of the development portions and research portions of the FSP use appropriate, separate treatments.

The assessment panel recommends that the FSP establish regular *informal* interaction with domain experts from both inside and outside the fusion community. Such domain experts should have a mutual interest in the progress and discovery being made by the FSP; however, they should not be direct participants in the FSP. To maintain the envisioned level of informality, interactions with this team should *not* be part of a review or constitute a formal advisory board.

Finally, the Proto-FSP Projects have significantly advanced the fusion community's understanding of multi-physics integration and modern HPC scientific software tools. They have explored different approaches and developed new capabilities. Nothing in this report should be regarded as an evaluation of the Proto-FSP Projects relative to their objectives, which we did not review. When the FSP is launched, it is important that there be one FSP team and that the three Proto-FSP teams contribute their knowledge, perspective, and developments to meet its greater challenges.

1. Introduction

The Fusion Simulation Program (FSP) is currently in its planning phase. The proposal for the FSP planning activity identified the need to assess the pre-existing "Proto-FSP" projects for identify best practices and lessons learned. These projects are

- The Center for Plasma Edge Simulation (CPES),
- The Center for Simulation of Wave Interactions with MHD (SWIM), and
- The Framework Application for Core-Edge Transport Simulation (FACETS) Center,

which are jointly supported by the DOE Offices for Fusion Energy Sciences (FES) and Advanced Scientific Computing Research (ASCR). The DOE program offices supported the need for the assessment in a letter from Dr. John Mandrekas (FES) and Dr. W. Polansky (ASCR), dated 16 June 2010, included in Appendix A. It asks that the assessment determine to what extent

a. the Proto-FSPs help contribute to establishing a foundation for the FSP; and
b. which elements of the Proto-FSPs would be most useful for the FSP.

The Proto-FSP Assessment Panel was formed to conduct the assessment and provide independent perspective on the issues. The six members of the committee are listed in Appendix B. The members include prominent experts on scientific simulation on leadership class computers and fusion from DOE national laboratories and a university.

The Assessment Panel met and discussed the issues on two conference calls. Background materials on the FSP and each of the Proto-FSPs were provided electronically and via the web. The background information included recent presentations and publications about each Proto-FSP's approach and accomplishments. The panel formulated a set of questions for the Proto-FSP projects, in Appendix C, to help identify the information needed in the assessment. The committee then met for four days at the Princeton Plasma Physics Laboratory on 30 August - 2 September. The meeting agenda, see Appendix D, included one day of presentations and discussion with each Proto-FSP team. The presenters were requested to address the DOE information request and to provide answers to the questions.

Each of the Proto-FSP Projects chose a fusion science modeling problem requiring integrated treatment of two or more physical processes previously treated separately. The problems chosen are significant and challenging, and are subsets of the issues to be addressed by the full FSP. All of the Proto-FSPs engaged experts from fusion science, applied mathematics, and computer science to help address their problems. Each of the projects prioritized their issues differently, choosing different multi-physics integration strategies and methods, and focusing their resources on different aspects of the overall problem. Often, the design choices were specific to the chosen problem and overall approach. Thus, the Proto-FSPs form a good set of prototyping R&D activities for the FSP, exploring a specific set of sub-cases.

The full FSP is a new program, significantly larger in every sense than the Proto-FSPs. The Assessment Panel chose to identify recommendations and best practices for the FSP without attribution, based on the combination of Proto-FSP experiences and our experience on other large projects. The recommendations regarding new technologies do not imply endorsement of any specific packages developed under the auspices of the Proto-FSPs. The assessment panel found value, insight, and useful results from each of the Proto-FSP Projects.

The remainder of this report is organized to address the recommendations for the FSP in the areas of project management, numerical algorithms, frameworks and software architecture, and physics components and users. Within each section we denote major recommendations using boldface.

2. Project Management

For a large tightly-integrated multi-physics simulation capability, project management must be well-organized to successfully coordinate the tasks and the teams implementing those tasks. **The assessment panel recognizes that a strong PI with breadth across the disciplines and clear vision is an essential component to a successful FSP.** The PI is responsible for ensuring proper coordination across the FSP effort and reporting to DOE program management. **We recommend that the PI adopt a decision-making process that is collaborative across stakeholders but not strictly democratic.** Each of the Proto-FSP multi-disciplinary teams are naturally structured around domain experts. **The assessment panel observed domain experts who both led their portion of the development effort *and* participated in the coordination of the overall project as a part of the management team; this model was effective and is recommended for the FSP.**

Each of the Proto-FSP Projects had as its central focus a critical science driver facing fusion science modeling. The choice of science driver had impact on project management, software architecture, algorithm development, and physics fidelity. It also directly affected the risk assessment and prioritization, leading to mission-driven scientific research and development. **The Assessment Panel *highly* recommends that the FSP use science drivers as its core organizing principle.** Considering the scale of the full FSP, a single science driver is unlikely to be sufficiently broad; however, six science drivers is likely to diffuse effort. The assessment panel recommends that the FSP planning process identify the two or three highest priority science drivers to focus the FSP at any given time.

Geographically-dispersed multi-institutional projects present particular challenges in coordinating teams. The FSP will need to acknowledge and anticipate the overhead in management and communication efforts of such distributed teams. **The organization of the project into well-defined tasks and work packages and associated working groups is recommended.** The FSP may also benefit from a full-time project manager, to be determined by the FSP management team. Yearly all-hands meetings of the entire project combined with quarterly meetings of working groups and frequent teleconferences will likely be required. All of the Proto-FSPs used these techniques to good effect. The FSP may need to explore additional collaborative technologies as the size of the teams increase (e.g., web-based video conferencing); the FSP itself should evaluate the efficacy of such solutions.

While multi-disciplinary teams present challenges as outlined above, all of the Proto-FSPs clearly recognized the benefits of assembling physicists, applied mathematicians, computer scientists, and software engineers to advance the state-of-the-art in plasma simulation science. Each Proto-FSP exhibited aspects of a "separation of expertise" where software and science people work on different parts of the project. In a large project such as FSP, this may lead to difficulties: a key concern for FSP should be the training of a generation of interdisciplinary scientists who are primarily physicists but are also interested in HPC and software engineering and vice versa. **Scientists should be encouraged and rewarded for interdisciplinary work; they require opportunities for career advancement for “writing code” rather than “writing papers.”**

There are certain teams that the assessment panel recommends: a frameworks team, key physics component teams, and a management team. All of these teams should be multi-disciplinary including applied mathematicians, computer scientists, physicists, and software engineers. The assessment panel also suggests that the FSP management team consider a science driver team for each science driver to address the mission focus.

The Proto-FSPs employed project management practices that were effective for the modest size of the development effort. Some of these practices will scale up to a larger project, such as the FSP; some will not. **The FSP needs to adopt a tailored approach with documented, managed, and understood practices that are commensurate with the risks of the development.**

All of the Proto-FSPs benefited from significant leveraging of other past and present efforts. This is, in part, because the plasma physics being explored is engaging to the scientific community. This is also, in part, motivated by the possibility of longer-term funding under the FSP. The FSP will be expected to fund staff at a level-of-effort more commensurate with their time commitments and actual contributions. **In particular, each component in the critical path of the FSP needs to be appropriately funded by the FSP to access the needed support and establish access to the source code.**

Each Proto-FSP managed risk through an informal process. Risks of software technology (e.g., commercial, recommended technologies) were studied and evaluated in an appropriate, if informal, manner. This seems reasonable considering the size and funding level of the Proto-FSP teams. **The assessment panel recommends that the FSP establish a systematic and appropriately tailored evaluation of risks (including, but not limited to, software, applied mathematics, and physics modeling) with clear communication of the decision that have been made across the FSP.** One of the significant risks to the FSP will be an orderly transition from utilizing the existing production codes (many of these are legacy codes) to new technologies, likely replacing those legacy tools as the FSP succeeds in its mission.

The culture of the fusion community does not currently embrace the Open Source software paradigm. The assessment panel observed clear advantages for those aspects of the Proto-FSPs that employed Open Source. Among the stated goals of the FSP is to deliver a community code. A view of a community code is one that allows scientists to explore a physics phenomenon to perform discovery science as well as planning for new and innovative experiments. This allows a model of a delivered software capability that will be used by the wider community but not modified. Another view of a community code is one that enables continued development of algorithms and physics models. This requires a paradigm that allows contributions to be incorporated back into the source code. This latter model presents significant challenges and benefits *and* will require a cultural change in the U.S. fusion community. **The assessment panel recommends to FSP program management that the risks of these models be evaluated and a definition of community code expectations be posed at the outset. Independent of these models, the assessment panel strongly recommends that the FSP program managers make open source licensing a formal requirement for the project.**

The leadership of the FSP will need to address the extent to which existing physics codes will be used unmodified in the context of multi-physics simulations. The assessment panel suggests that the FSP may benefit from a cultural shift from independent production codes that are then used within a framework to a concept of physics components that have applicability as independent codes.

Utilization of non-Open Source elements in the FSP will require their verification and validation. In addition, the FSP will not be able to extend or refactor such code, and may need to negotiate its further distribution in a FSP release. **For these reasons, we recommend restricting use of non-Open Source code to well documented, well supported libraries and separate tools.**

All Proto-FSPs identified embedded development (e.g., with the base program or physics community) as a highly effective mode for collaboration and necessary for success, both to create new models and for adoption of the FSP by the community. Embedded liaisons/developers requires sufficient understanding of at least two domains of expertise (e.g., physics and framework technology) to communicate effectively. These liaisons should be directly supported for this work through the FSP.

3. Numerical Algorithms

All teams discussed the need for more implicit coupling algorithms but are just in the beginning of developing cross-component implicit schemes. All groups have seen instabilities due to explicit coupling. Picard iteration, a more sophisticated quasi-Newton, and full Newton technique were deployed. All teams explored implicit schemes *within* particular components. For dynamics with a larger time scale disparity this effect will be important to address. All teams have explored different techniques and all will likely need to be part of the tool set available to a full FSP effort. **Advanced implicit algorithms will be needed for different aspects of the multi-physics integration; such as Newton, Quasi-Newton and Picard schemes, as well as Jacobian-Free Newton Krylov approaches.** To use more advanced techniques, achieving improved performance, the FSP must be ready to significantly modify or re-design the component interfaces and implementations.

This panel also feels that advanced time-stepping is an important feature for stable time evolution for complicated non-linear multi-component physics. This will be a challenging development goal. Such time-step controllers are actively used in other projects, including climate and HEDP models and these other communities should be surveyed for suitable technology. **We expect that advanced time-advance algorithms will be required for different aspects of the multi-physics integration; candidates include adaptive and parallel-in-time algorithms. It is important that this research continue, and that the FSP framework be flexible enough to support any or all of these.**

Liaisons, or mediators, between a domain expert and the relevant development team (e.g., framework team or physics component team) can be crucial to successfully integrate algorithm development. The Proto-FSPs observed instances in which domain experts may be more efficient with a good liaison with complementary expertise. **We recommend that these cross-discipline liaisons be a standard practice for the FSP.**

All teams made use of spatial and temporal and statistical self-convergence studies for verification. Some teams also made use of code comparisons against analytic and semi-analytic (i.e., independently error-controlled) solutions for verification studies. Some teams also made use of the method of manufactured solutions. **All of these techniques have a utility and are recommended practices for the FSP.**

By using verification test problems, teams were able to demonstrate the spatial and temporal order of accuracy of the physics algorithms being employed. Maintaining second- or even first-order convergence can be challenging in multi-physics simulations. There are few analytic and semi-analytic solutions for such coupled physics problems. The FSP (and many other scientific communities) would benefit from an increase in research in this area.

Some teams are using cross-code comparisons and refer to them as "verification". This activity provides valuable information under certain conditions; however, oftentimes the reference codes are not verifiable themselves. The problem of cross-code comparisons is ensuring they are modeling identical problems: these codes tend to vary in what effects they include and how. Without access to the source code it is very hard to simulate identical physics processes and problems. One needs to model the same problem; ideally the reference solution is arrived at using a distinct error-controlled numerical technique.The panel is in agreement on the utility of these activities. The DoD definition for verification is "The process of determining that a model or simulation implementation and its associated data accurately represent the developer's conceptual description and specifications. [DoDI 5000.61, December 9, 2009]." This is sometimes referred to as "solving the equations right." **The assessment panel recommends that the FSP identify a common set of terminology to be used to facilitate communication, in particular identifying benchmarking and verification as distinct activities.**

Some of the Proto-FSPs have begun efforts on sensitivity analysis; the Proto-FSPs have not yet had the opportunity to explore uncertainty quantification (UQ). Sensitivity analysis can be used more extensively to guide and prioritize development (e.g., physics fidelity, algorithm accuracy). **The assessment panel recommends that the FSP incorporate use of sensitivity analysis throughout the development cycle.** UQ is an area of active research in several communities, including climate science and stockpile stewardship. The FSP will rely on their expertise and advances in UQ science. The assessment panel recommends that the FSP engage these communities early in the development cycle to gather potential requirements and best practices. **The assessment panel recommends that the FSP have a systematic plan for quantifying uncertainties in their simulations so that their significance can be understood.**

Documentation for algorithms should not be left only to the publications and conferences. It is a critical element of communication with users and other team members. **We recommend that the models and algorithms employed be documented as part of the component distribution or release process.**

4. Frameworks and Software Engineering

The use of a framework with well-defined interfaces for components is widely recognized to be beneficial to the scientific process. There is a consensus that the FSP software will be managed as a set of relatively distinct physics modules, the "Physics Composition Software" (which we will informally call "components"), and glue code, the "Task Composition Software" (which we will informally call a "framework"). No particular properties or functionalities of components or frameworks should be inferred from the computer science definitions of these terms in this document. There also appears to be a consensus that components do not directly call other components; rather, the communication is handled through the framework. There is not a consensus on whether all the components need to be compiled together or whether separate executables can be used. There was a more specific observation that when components needed functional access to external capabilities all three teams employed a library or common executable approach. That is, these external capabilities were not moderated through the component interfaces. The decision process was based on functional granularity, independence of parallel execution model, and whether the functionality required state. **We recommend the FSP remain cognizant of these issues when making design decisions about where certain functionality belongs, and be flexible to changing the locations as the project progresses. For example, functionality previously coded within several components may be migrated to the framework.**

The FSP should remain flexible to redesigning and refactoring its systems and requirements over the lifetime of the program and develop a managed (but not overly bureaucratic) system for evaluating and adopting changes. Enough support should be provided to ensure that each component team has the resources to adapt its components to changes in the system.

Based on the experiences of the Proto-FSPs and our experiences, we provide the following recommendations for frameworks.

The FSP should develop and maintain a single framework system. This framework should be lightweight. Any external libraries referenced should be optional. Since the fusion community is beginning to use unified frameworks, we recommend starting slowly, building upon existing efforts, while remaining open to the benefits of a more cohesive framework with standard state variables, import, export, regridding, and timestepping methods. The dangers of developing and maintaining multiple frameworks include the dispersal of expertise across the frameworks and that component developers will have to devote substantial resources to comply with multiple frameworks or will program to the lowest common denominator.

Informally, in scientific computations workflow can be defined as everything that happens outside the large parallel components. It encapsulates the steps required to execute particular simulations. **The FSP framework should either include or work with both scripting (for example, python) and formal workflow (for example, Kepler) control of the components.** We provide no recommendation on whether the framework should support or allow more than one scripting or formal workflow mechanism. We understand both these approaches may not be fully functional for 2 or 3 years. **The formal workflow system should not be necessary for using the framework.** Workflows, especially those capable of capturing and storing provenance information, are essential to being able to reproduce simulation experiments. Reproducibility of simulations must be considered an essential element of an FSP. There is not yet consensus on the use of GUI- or web service-driven formal workflows, and a large part of the community clearly wishes to work "close to the metal," being able to intervene directly in source code and scripts without higher levels of indirection. We recommend the use of a framework that allows unfettered access to the underlying scripts and source. But we also believe that the capture of provenance information and the ability to exactly reproduce a prior run should be essential elements of the FSP.

The framework should be recursive allowing instantiations of the framework to serve as components in another framework. There are multiple drivers for this recommendation, for example, uncertainty quantification will require this capability.

The framework should contain a "data exchanger" interface that manages the movement of data between components. For both data exchange and I/O we feel the requirements and options are too immature to recommend particular mechanisms at the start of the FSP. Rather, we recommend that the FSP make an early systematic study of the various options for both data exchange and I/O before selecting the model, including input from other multiphysics simulation project~~ion~~s, (and possibly develop prototypes of new systems or extentions of previous systems). The evaluation should include both low-dimensional data (scalar at a surface) and high-dimensional data (vector quantities across a well-resolved 3D volume). Interpolation (any translations between representations) and regridding could be part of this as well. The evaluation needs to include dataset sizes that extend to the projections that the FSP anticipates (e.g., tera- or petabyte datasets), which may not have been encountered by the Proto-FSPs. Also important is a systematic approach to capturing the provenance of the I/O data sets and the inclusion of experimental data. To allow progress in other areas during the evaluation, the FSP should initially select prototype systems for data exchange and I/O and then switch to the designed system.

We note that current APIs for I/O (such as HDF5) are similar to those for data exchange. Therefore **we recommend that the FSP evaluate the efficacy of a unified interface for both**

data exchange and I/O. This unified mechanism could also be used for provenance, allowing rollback and replaying of component interactions and generating restart files. Note that this does not mean we are recommending any particular current technology to achieve this.

Different people have different ways of developing and using software. The FSP should allow as many as possible and not erect unneeded barriers to certain approaches, for example, imposing an IDE on all developers or a single model for workflow. The FSP should, as much as practical, allow components to be written in a range of common languages, including C, C++, Fortran, and python. We suggest discouraging the use of languages that are not widely used in the Open Source community.

We recommend that the framework developed allow incremental adoption of its technology since this means it will be far more likely to be adopted by a wider community. A goal for a highly-effective framework environment is that **it should provide incentives to the domain experts to accomplish their research goals within the framework rather than by bypassing the framework**.

With regard to components we have the following recommendations.

FSP should develop a set of requirements on interfaces that each component must satisfy. This set should be as minimal as possible while requiring enough functionality to make the approach practical.

Software training and documentation on making components from scratch and from current codes must be available within a few months of the initiation of the FSP, with an understanding that the details will change as the design of the framework evolves.

All components need to be maintained in a source control system. Since the FSP development will occur over a long period, individual teams should be allowed to explore new approaches. We do not recommend requiring that a single source control system be used by all components. We recommend the project provide a single website that provides access to all the software components, this does not mean that all the software need be hosted at a single site. **Similarly, all documentation should be maintained in a version control system and made available at a centralized web site** (even if component documentation is originated at multiple institutions)**.**

Each component must have a test suite that is run regularly and after any changes, that tests both internal functioning of the component and its usage in the framework. The framework, of course, also needs extensive regular testing. Purely in terms of return on investment, the FSP must insist on testing before launching large-scale runs. A convenient rule-of-thumb is that a core-hour on one of the LCFs costs 15-20 cents; thus a one-hour run on 200,000 cores costs $30-40,000. This is sufficient justification for automated unit testing and system testing as an essential component of software development. We expect that the framework team will maintain a flexible testing environment for use by the component teams. Individual teams should be allowed to innovate beyond this.

Each component must have some level of portability, from laptops to LCFs, on various systems as determined by the FSP; enough resources must be provided to the component teams to maintain this portability; this includes the fact that the FSP must ensure that each component team has access to appropriate hardware for testing. Component teams must insure portability of any external libraries that they rely on.

Many software development and quality assurance processes and practices are used in scientific computing. As stated, the assessment panel recommends pervasive use of software configuration management and regression testing. Beyond this, effective practices will need to be tailored to the development teams and the risk associated with the effort. **The assessment panel recommends that the FSP identify and document their practices in a software development and quality assurance plan.**

The choice of how much functionality to implement within a single component, as opposed to providing several components that are used together, will sometimes be difficult. The advantage of several components is that the same functionality can be immediately reused by other components, while if it is embedded within a component the source code may need to be extracted and copied into another component. Also, separation into several components enables support and development at a smaller scale size, which can increasing efficiency. The difficulty is if two pieces of functionality require extremely tight coupling for high performance they cannot practically be in different components. The FSP should remain cognizant of this issue and carefully design the extent of components.

When developing new components and evaluating existing physics applications for development as components, we expect that scalability, flexibility, portability, and dynamic load balancing of the component will be critical aspects to realize the goals of the FSP.

The framework should support gathering and organizing performance information from each component. Similarly, each high-quality component should be able to provide its performance data to the framework.

The FSP needs to maintain awareness of the evolution of HPC architectures and programming models, adapting framework and new component development as needed.

We recommend the incorporation of a FSP specific web-based portal for job definition, job submission, job monitoring, and post-job analysis. This should be designed so that multiple portals can be run by different groups. We recommend that this portal use only open-source technologies (for example, HTML 5.0). This portal should not be necessary for using the framework. This same portal could be used for accessing data sets, both from simulations and experiments.

5. Physics and User Elements

The three Proto-FSP projects chose interesting and important physics topics, requiring significant integration of models of separate physical processes. These topics form a subset of the topics the FSP will address. All three projects built upon legacy codes to model some of the needed processes, in order to get results in a timely manner. This also allowed them to make use of previous work validating the chosen models. In order to integrate across the chosen multiple physics domains, all three projects found that they needed to modify the legacy models (sometimes significantly) or upgrade them with new algorithms. **The FSP should similarly expect that significant modifications will be required to incorporate and integrate legacy models.**

The most novel physics results from the Proto-FSPs are coming from new physics models, incorporating new physics characteristics and approaches. The Proto-FSPs are also exploring new insights from first-of-a-kind multi-physics simulations. All of the Proto-FSPs found progress toward increasingly integrated multi-physics simulation capability to be proceeding more slowly than planned, due to numerical instabilities and other issues. In the opinion of the assessment panel, this is dominantly the result of multi-physics coupling being fundamentally difficult. **The FSP will need to analyze, design, and manage the physics-integration numerical interfaces carefully.** Iteration and improvement of the physics models and the numerical interface between them will be key to achieving the FSP goal of an authoritative model.

The Proto-FSPs have chosen different physics decompositions in designing similar physics components and this clearly impacts the required communication and potentially available fidelity. Multiple component structures must be available and usable simultaneously, to test the advantages of different approaches. **The FSP framework must be flexible enough to support multiple componentization choices.** The appropriate choice of physics component boundaries should be analyzed as part of the interface design, to ensure successful integration with other components.

The Proto-FSPs have begun validating their integrated results by limited comparisons with experimental measurements. The results are qualitative, with unquantified uncertainties. To focus its efforts, the FSP should consider adopting a clear definition for validation, such as the DoD definition "The process of determining the degree to which a *model and its associated data* are an accurate representation of the real world from the perspective of the intended uses of the model." In addition, **the FSP should establish a clear validation process and standards, as part of a larger software quality plan**. Such a plan could target existing standards and principles, such as those developed in the ASC Program, see *"ASCI Software Quality Engineering: Goals, Principles, and Guidelines"* (DOE/DP/ASC-SQE-2000-FDRFT-VERS2), and related lab-specific documents. **As part of such a plan, we recommend that the FSP define reference experiments for qualifying models that exercise the integrated system not just individual components.**

The Proto-FSPs have already observed that integration and use of physics components outside their range of validity can produce unreliable integrated predictions, depending on the overall sensitivity to those unreliable models. **The FSP should include a process for validating and qualifying individual models and physics components, and determining their range of validity.** While clearly not predictive, unreliable models can be useful to prioritize and guide multi-physics development.

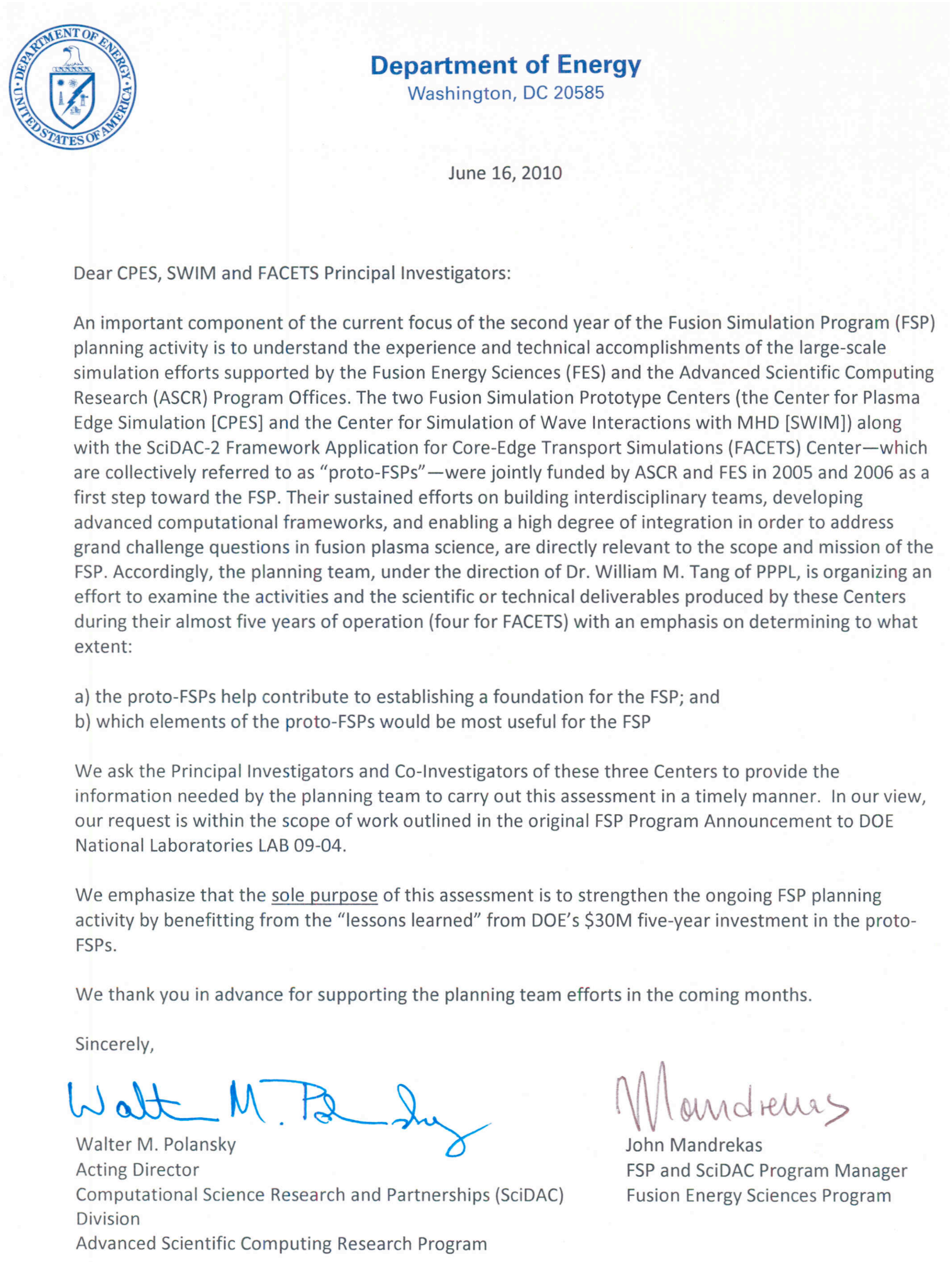
The Proto-FSPs are just starting to have significant external users. However they are already very important for providing feedback to the development team, validating models and choices, and providing real use cases. All three groups developed visualization tools and tools for overseeing and managing runs. However, tools to easily prepare runs are at an early stage of development.

In order to attract external users, the tools need to provide new capabilities and the opportunity for new scientific discovery. This may require a more mature tool than is available early in a project's development. **We recommend that the FSP include internal users, who ensure early testing and exploration of the physics integration use cases.** This may be satisfied by dedicated usage of the tools by some of the developers. **In addition, we recommend that the FSP directly support a set of early adopters, attempting to produce science results, to provide feedback on the learning curve and needed user tool sets.**

The user tools for a powerful scientific modeling system must support a range of users, including intensive "power users" and infrequent or neophyte users. **The FSP user tools must be sufficiently flexible to support the full range of users engaging in discovery science.** It should include interfaces to scripting languages to support high volume production, as well as GUI-tools to enable easy routine runs by occasional users.

The Proto-FSPs are evolving rapidly, and are focused on exploring their chosen topics. A natural consequence is that documentation is deferred. **The FSP must ensure that adequate documentation is prepared as an integral part of development.** This is key to attracting and maintaining a significant user base, and to fulfilling the FSPs role as a tool for the whole fusion community.

Appendix A



Department of Energy
Washington, DC 20585

June 16, 2010

Dear CPES, SWIM and FACETS Principal Investigators:

An important component of the current focus of the second year of the Fusion Simulation Program (FSP) planning activity is to understand the experience and technical accomplishments of the large-scale simulation efforts supported by the Fusion Energy Sciences (FES) and the Advanced Scientific Computing Research (ASCR) Program Offices. The two Fusion Simulation Prototype Centers (the Center for Plasma Edge Simulation [CPES] and the Center for Simulation of Wave Interactions with MHD [SWIM]) along with the SciDAC-2 Framework Application for Core-Edge Transport Simulations (FACETS) Center—which are collectively referred to as "proto-FSPs"—were jointly funded by ASCR and FES in 2005 and 2006 as a first step toward the FSP. Their sustained efforts on building interdisciplinary teams, developing advanced computational frameworks, and enabling a high degree of integration in order to address grand challenge questions in fusion plasma science, are directly relevant to the scope and mission of the FSP. Accordingly, the planning team, under the direction of Dr. William M. Tang of PPPL, is organizing an effort to examine the activities and the scientific or technical deliverables produced by these Centers during their almost five years of operation (four for FACETS) with an emphasis on determining to what extent:

a) the proto-FSPs help contribute to establishing a foundation for the FSP; and
b) which elements of the proto-FSPs would be most useful for the FSP

We ask the Principal Investigators and Co-Investigators of these three Centers to provide the information needed by the planning team to carry out this assessment in a timely manner. In our view, our request is within the scope of work outlined in the original FSP Program Announcement to DOE National Laboratories LAB 09-04.

We emphasize that the sole purpose of this assessment is to strengthen the ongoing FSP planning activity by benefitting from the "lessons learned" from DOE's $30M five-year investment in the proto-FSPs.

We thank you in advance for supporting the planning team efforts in the coming months.

Sincerely,

Walt M.

Walter M. Polansky
Acting Director
Computational Science Research and Partnerships (SciDAC)
Division
Advanced Scientific Computing Research Program

Mandrekas

John Mandrekas
FSP and SciDAC Program Manager
Fusion Energy Sciences Program

Appendix B

Panel Membership

Dr. Venkatramani Balaji, NOAA / Geophysical Fluid Dynamics Lab.

Dr. Barry Smith, Argonne National Lab.

Dr. Brian Van Straalen, Lawrence Berkeley National Lab.

Prof. Priya Vashishta, University of Southern California

Dr. Michael Zarnstorff (Chair), Princeton Plasma Physics Lab.

Dr. Michael Zika, Lawrence Livermore National Lab.

Appendix C

Questions for Proto-FSP Teams

- For each of the following foundation elements
    - What approaches worked well? What would you consider a best practice?
    - What approaches worked poorly?
    - What approaches would you recommend as appropriate for the full FSP?

    Project & team management, including risk management
    Physics scope & fidelity
    Algorithm choice and development
    Software architecture, including change control
    Quality assurance
    Uncertainty quantification
    User support
    Verification & validation
    Documentation, internal and external
    Component lifecycle process

    Are there additional foundation elements? If so, please discuss your approaches (a-c) to them.

- What specific techniques are you using to insure the stability and accuracy of your coupling schemes? How do you decide what accuracy is sufficient and if higher order-accurate methods are needed (in time or space)? Do you have methods to test different coupling schemes and verify them using your actual simulation codes?

- Was an external user group and community established for the project? Was the project used to gain new scientific understanding? What were the important aspects in how this developed? How is your project software released and licensed? How did the user community influence your decisions and development?

- What aspects or components of your Proto-FSP do you consider most useful for the FSP? Why?

- In the presentations on 10 August, there was discussion of mingling the Proto-FSPs. What aspects or components in the other Proto-FSPs are most useful? What challenges do you foresee in cross-integration? What is your perspective on the other groups' approaches?

Appendix D

Proto-FSP Assessment Agenda
30 Aug. – 3 Sept. 2010
PPPL, Princeton, NJ

Monday, 30 Aug., Room B318

8:30 am	Panel Discussion
9:00 am	SWIM presentations and discussion
D. Batchelor	FSP Background
D. Batchelor	Rationale and Physics Background for SWIM
D. Bernholdt	Overview of SWIM's Computational Approach
S. Jardin	NSTX MHD
S. Kruger	Slow MHD
D. Batchelor	ITER Simulations, Parareal, and others
D. Batchelor	Answers to Panel Questions
5:00 pm	Panel Discussion

Tuesday, 31 Aug., Room B318

9:00 am	CPES presentations and discussion
C.S. Chang	Overview of the Physics-Design Principles at Proto-FSP CPES
S.A. Klasky	EFFIS
M. Parashar	Summary and Conclusions
C.S. Chang	CPES Answers to the Questions for Proto-FSP Teams
5:00 pm	Panel Discussion

Wenesday, 1 Sept., Room B318

9:00 am	FACETS presentations and discussion
J.R. Cary	FACETS Overview
A. Hakim	Overview of the FACETS Framework
L. McInnes	Applied Math Algorithms in FACETS
T. Epperly	Software Engineering
S. Kruger	Physics Studies with the FACETS Framework
J.R. Cary	FACETS Concluding Remarks
5:00 pm	Panel Discussion

Thursday, 2 Sept., Director's Conference Room, B331

9 – 5pm	Panel Discussions and Writing

Friday, 3 Sept., Conference Call

1 – 5pm	Panel Discussions and Writing