SFA Modernization Partner

United States Department of Education

Student Financial Assistance

ITA Release 2.0
Strategic Assessment

Task Order #69

Deliverable # 69.1.1

February 1, 2002

Table of Contents

1 Executive Summary 5

1.1 INTRODUCTION / OBJECTIVES 5

1.2 APPROACH 5

1.3 INTEGRATED TECHNICAL ARCHITECTURE RELEASE 2.0 OVERVIEW 5

1.4 INTEGRATED TECHNICAL ARCHITECTURE RELEASE 3.0 OVERVIEW 7

2 Integrated Technical Architecture (ITA) 8

2.1 ITA OVERVIEW 8

2.2 ITA PRODUCTS AND UPGRADES 8

2.3 ITA RELEASE 2.0 9

3 Assessment Summary 9

3.1 STRENGTHS 9

3.1.1 Environment Support 9

3.1.2 Products Support 10

3.1.3 Reusable Common Services (RCS) 10

3.2 WEAKNESSES 10

3.2.1 Communication 10

3.2.2 Use of RCS 10

3.2.3 Enhancements to RCS 10

3.2.4 Additional Services Requested 11

3.2.5 Automated Environment Building 11

3.3 PRIORITIZED RECOMMENDATIONS 11

3.4 SCOPE OF ITA RELEASE 3.0 12

3.4.1 Reusable Common Services *12*

3.4.1.1 Configuration Service 12

3.4.1.2 Web Conversation Framework 13

3.4.1.3 Scheduler Framework 13

3.4.1.4 Session Management 14

3.4.1.5 JSP Tag Library 14

3.4.1.6 XML Helper 14

3.4.1.7 File Transfer Protocol Service 15

3.4.1.8 Object Pooling 15

3.4.1.9 Technologies Enabling Web Services 15

3.4.1.10 Portlets 17

3.4.2 ITA Core Services *18*

3.4.3 ITA Software *18*

3.4.4 Performance Enhancement Services *19*

3.4.5 SFA Technical Architecture Practices *19*

3.4.6 ITA Scripting *19*

3.4.7 ITA Workshops *19*

3.4.8 ITA Product Specialty Support *20*

3.4.9 Independent Verification & Validation *20*

4 SFA Projects 21

4.1 OVERVIEW 21

4.2 ECAMPUS BASED (ECB) 21

4.2.1 Functional Description *21*

4.2.2 Status *22*

4.2.3 ITA 2.0 Services Used *22*

4.2.4 Enhancements/Upgrades *23*

4.2.5 Potential ITA 3.0 Services *23*

4.2.6 Schedule *24*

4.3 COMMON ORIGINATION AND DISBURSEMENT (COD) 24

4.3.1 Functional Description *24*

4.3.2 Status *25*

4.3.3 ITA 2.0 Services Used *25*

4.3.4 Enhancements/Upgrades *26*

4.3.5 Schedule *26*

4.4 ENTERPRISE ARCHITECTURE INTEGRATION (EAI) 27

4.4.1 Functional Description *27*

4.4.2 Status *27*

4.4.3 ITA 2.0 Services Used *27*

4.4.4 Enhancements/Upgrades *28*

4.4.5 Schedule *28*

4.5 ENTERPRISE INFORMATION PORTAL (EIP) 28

4.5.1 Functional Description *28*

4.5.2 ITA 2.0 Services Used *29*

4.5.3 Enhancements/Upgrades *29*

4.5.4 Potential ITA 3.0 Services *29*

4.5.5 Schedule *30*

4.6 FREE APPLICATION FOR FEDERAL STUDENT AID (FAFSA) 30

4.6.1 Functional Description *30*

4.6.2 Status *31*

4.6.3 ITA 2.0 Services Used 31

4.6.4 Enhancements/Upgrades 32

4.6.5 Potential ITA 3.0 Services 32

4.6.6 Schedule 33

4.7 FINANCIAL PARTNERS (FP) DATA MART 33

4.7.1 Functional Description 33

4.7.2 Status 34

4.7.3 ITA 2.0 Services Used 34

4.7.4 Enhancements/Upgrades 34

4.7.5 Potential ITA 3.0 Services 34

4.7.6 Schedule 34

4.8 INFORMATION FOR FINANCIAL AID PROFESSIONALS (IFAP) 34

4.8.1 Functional Description 34

4.8.2 Status 35

4.8.3 ITA 2.0 Services Used 35

4.8.4 Potential ITA 3.0 Services 35

4.8.5 Schedule 35

4.9 OPERATIONS 36

4.9.1 Functional Description 36

4.9.2 Status 36

4.9.3 ITA 2.0 Services Used 36

4.9.4 Potential ITA 3.0 Services 36

5 Appendices 38

1 Executive Summary

1.1 Introduction/Objectives

The *Integrated Technical Architecture (ITA) Release 2.0 Strategic Assessment* analyzes the ITA R2.0 architecture and identifies the strengths and improvement areas for the deployed architecture. This analysis provides a thorough review of all of the SFA applications that are deployed with or are planning to use the ITA architecture. Based on the analysis and input from the application teams, the document also identifies and prioritizes ITA enhancement areas for the ITA Release 3.0 (R3.0) architecture.

This document provides a summary of the Student Financial Assistance's (SFA) projects, how they have implemented ITA R2.0, strengths, areas for improvement, prioritized enhancement requests and new services for ITA R3.0, and schedules. The ITA R2.0 Strategic Assessment is divided into the following sections:

- Section 1 provides a high level overview of the strategic assessment, including the projects that were assessed, an overview of ITA R2.0, strengths and weaknesses of ITA R2.0, enhancements, and new services that will be provided in ITA R3.0.
- Section 2 provides a detailed overview of the ITA R2.0 architecture, products, enhancements, upgrades, and new common services for ITA R3.0.
- Section 3 provides an analysis of the SFA applications.
- Section 4 provides an assessment summary with strengths, weaknesses, and prioritized recommendations.
- Appendices include the Technical Description for SFA projects, Application Services Matrix, and the Strategic Assessment Questionnaire.

1.2 Approach

The ITA team collected information for the Strategic Assessment through interviews with application team leads and members of the ITA R2.0 team. Additional information on SFA's projects and ITA R2.0 were collected through the review of existing application documentation.

The ITA team interviewed the application team leads and posed a common set of questions (Appendix IV) regarding the applications' functional and technical descriptions, ITA R2.0 services used, strengths and weaknesses of ITA R2.0, and potential ITA R3.0 services.

1.3 Integrated Technical Architecture Release 2.0 Overview

ITA R2.0 provided SFA with the core functionality to support the deployment of the initial SFA applications. ITA R2.0 provided a standardized, reusable infrastructure for enabling business capabilities. Those existing business capabilities are: Standard product framework, build and setup of environments, SME support, troubleshooting and bug fixes, Reusable Component

Services (RCS) Release 1, and ITA recommended coding standards and best practices. Future releases will build on R2.0, adding new technologies, as required.

The ITA framework consists of four domains:

- Environment Support
  - Environment support consists of environment building, configuration setup, maintenance and trouble shooting for applications and environments.
- Products support
  - Products support helps application teams install, configure, and trouble shoot the ITA recommended product set including IBM HTTP Server (IHS), IBM WebSphere Application Server (WebSphere), IBM Network Dispatcher, Autonomy, Interwoven, and Viador.
- Reusable Common Services (RCS)
  - A set of ITA developed and tested architectural services that are common to web application development. The services are designed for reusability across different applications. The RCS are constructed on the Java 2 Enterprise Edition (J2EE) platform. Portlets, which are reusable components for portal applications, are part of RCS.
- Subject Matter Expertise (SME)
  - The ITA provides assistance and recommendations for application teams on a variety of subjects such as developing Java based applications and coding standards.

SFA Projects

This document will assess the usage of ITA by the following SFA projects:

- eCampus Based release 2 (eCB)
- Common Origination and Disbursement (COD)
- Enterprise Architecture Integration (EAI)
- Enterprise Information Portal (EIP)
- Free Application for Federal Student Aid (FAFSA)
- Financial Partner's Data Mart (FP Data Mart)
- Information for Financial Aid Professional (IFAP)
- Operations Team
- Portals

- Electronic Audits and Financial Statements (EAFS)
- Debt Management and Collection Systems (DMCS)

1.4 Integrated Technical Architecture Release 3.0 Overview

Based on analysis of ITA Release 2.0, several enhancements were identified. The key enhancement areas identified in ITA R2.0 are:

- Communication
- Use of ITA core service
- RCS enhancements
- Additional reusable common services and portlets
- Automated environment building

The analysis also revealed a need to upgrade software such as: WebSphere Application Server, IBM HTTP Server (IHS), Network Dispatcher, Autonomy, Interwoven, and Informatica.

ITA Release 3.0 will introduce a number of new services which include: Operation Infrastructure support, additions to Coding Standards (Java & SQL), Configuration Framework, Web Conversation Framework, Scheduler Framework, Session Framework, JSP Tag Library Framework, XML Helper, File Transfer Protocol Service, and Object Pooling Framework, Technologies Enabling Web Services which include: Simple Object Access Protocol (SOAP) Framework, Universal Description Discovery and Integration (UDDI) Framework, and Portlets.

Finally, the ITA Core Services will continue with modifications and enhancements to these existing services: ITA Software, Performance Enhancement Services, SFA Technical Architecture Practices, ITA Scripting, ITA Workshops which are a new addition to the ITA communication strategy, ITA Product Specialty Support, and Independent Verification and Validation.

2 Integrated Technical Architecture (ITA)

2.1 ITA Overview

ITA provides an infrastructure that facilitates the integration of disparate data resources, business services, and process automation. ITA is based on four core architecture domains within the SFA technical environment, which are targeted at reducing stovepipe systems, islands of technology, and the need for customized point-to-point system interfaces. The four categories are:

- Environment Support
- Product Support
- Reusable Common Services
- Subject Matter Expertise

ITA has evolved over time to become the center of infrastructure support. In the environment support domain, ITA provides environment build, test and maintenance for applications under development and production. In the product support domain, ITA is the point of contact for product maintenance on SFA standard products such as IBM HTTP Server, Network Dispatcher, IBM WebSphere, Autonomy, Vaidor, and Interwoven. For RCS, ITA constructed common infrastructure frameworks such as email, login, exception handling, search, component factory, and persistence for application teams to use in order to shorten development time. On the SME front, ITA provides support for application teams on a wide range of subjects, such as: Application architecture expertise, product expertise, Java expertise, and capacity planning.

In combination, the four core ITA domains provide the necessary infrastructure to implement a service-oriented architecture. The four domains overlap to provide seamless integration between domain services. For instance, reusable common services are built on top of the SFA environment and are supported by the ITA team.

2.2 ITA Products and Upgrades

One of the main goals of the ITA is to standardize hardware and software products across SFA. In order to ensure that the standard framework is current, the ITA will periodically upgrade the version of software supported. The following table summarizes the products and the versions supported by ITA 2.0 and ITA 3.0.

Function	Product	ITA 2.0	ITA 3.0
HTTP Server	IBM HTTP Server (IHS)	v. 1.3.6 v. 1.3.12.2	v. 1.3.12.2 v. 1.3.19 (TBD)
Java Application Server	WebSphere Advanced Application Server	v. 3.02 v. 3.5	v. 3.5 v. 4.0 (TBD)

Search Engine	Autonomy	v. 2.1	v. 2.1
Cluster Load Balance and Failover	eNetwork Dispatcher	v. 2.1.2.0 v. 3.0	v. 3.0
Content Deployment and Management	Interwoven TeamSite	v. 4.2.1 v. 5.0.1	v. 5.0.1
Development Tools	Visual Age for Java	v. 3.5	v. 3.5 v. 4.0 (TBD)
Database	Oracle 8i	v. 8.1.6 v. 8.1.7	v. 9i

2.3 ITA Release 2.0

In release 2.0, the ITA continues to provide application teams with a core set of services -- product support, SME support, capacity planning, and environment build and configuration. In ITA release 2.0, RCS Release 1.0 is also added to ITA services.

3 Assessment Summary

The assessment summary section provides information on the strengths and weaknesses of Release 2.0 of the ITA architecture. This section also includes a section with recommendations that specifically address the weaknesses identified in ITA R2.0. The weakness, recommendation, and priority level are listed in a matrix (Section 3.3).

3.1 Strengths

The primary strengths of ITA R2.0 architecture are the Environment Support, SME and RCS Frameworks.

3.1.1 Environment Support

ITA R2.0 continues to provide environment build out, setup and bug fixes for applications. The ITA team leverages its SME in products such as WebSphere and IHS to more efficiently build test, development and production environments that meet the requirements set by the application teams. ITA also helps to configure applications to run on the environments when needed. All the application teams that were interviewed valued the services provided by the ITA team.

3.1.2 Products Support

The ITA 2.0 supports its recommended product set. It is the center for installation, configuration and trouble shooting of products like Interwoven TeamSite, Viador, Autonomy, IHS and WebSphere.

3.1.3 Reusable Common Services (RCS)

ITA's RCS received commendable reviews from application teams. The RCS were well received by application teams, whose feedback confirmed that the ITA RCS were well designed, constructed, and tested. RCS provides a common set of software infrastructures that allows developers to build applications within a shorter development cycle with high quality of infrastructure code. As new applications are released more RCS components will be utilized by these applications. The ITA will continue rolling out new RCS components such as Web Conversation Framework, XML Helper, Scheduler, and Session Management Framework.

3.2 Weaknesses

The key weaknesses identified in ITA R2.0 are insufficient communication, insufficient number of teams taking advantage of the ITA framework and core services, and the need for additional ITA services.

3.2.1 Communication

Communication among ITA and application teams has improved since ITA 1.0. Specifically, ITA 2.0 implemented the issue submission process, as issue submission and tracking were identified as a weakness in ITA 1.0 strategic assessment. There remains scope for improving communication in ITA 3.0. The ITA will also hold ITA workshops to raise awareness of the application teams of its services and reusable components services. By doing so, application teams can learn about all the services that ITA offers and utilize them in the development and production stages.

3.2.2 Use of RCS

In the interviews with application and development teams, the ITA team found that some of the teams did not implement RCS components in their applications because the teams did not know about RCS early in the project life cycle. ITA will also strive to communicate its core services to both application and development teams in advance. Doing so will allow teams to focus on application specific development rather than on the infrastructure services that are provided by ITA RCS.

3.2.3 Enhancements to RCS

After rolling out RCS, additional functionalities were requested by application teams. For instance, there was a request to enhance the Logging and Email frameworks. New functionality such as adding a scheduler for sending out mass emails has also been requested.

3.2.4 Additional Services Requested

The success of RCS led to frequent requests by application teams for additional common services. Some of the services that the application teams would like to have are: Web Conversation Framework, XML Helper, Scheduler Framework, and Configuration Service. The ITA team is currently evaluating each request and will roll out the new services in RCS for ITA R3.

3.2.5 Automated Environment Building

In the course of ITA 2.0 , the ITA team identified that a more efficient way of environment building was needed. ITA 2.0 environment services provided application teams with support on building and configuring different environments. Environment build out and configuration is a long and repetitive process. Shell scripting can potentially be used to carry out the tasks. Automated Scripting will allow application teams to setup their own environments and provide consistency across test, development and production environments.

3.3 Prioritized Recommendations

The following table summarizes the weaknesses and recommendations as well as assigning a priority to the area.

Priority	Weakness	Recommendation
High	Communication	Workshop to communicate ITA services offered. Notify scheduled outage to application teams.
High	Use of RCS	Communicate new services to application teams in advance.
High	Additional Services Requested	RCS Release 2
Medium	Enhancements to RCS Release 1	Provide enhancements to RCS frameworks based on agreed upon requirements.
Low	Automated Environment Building	Shell scripting for environment building and configuration.

3.4 Scope of ITA Release 3.0

In keeping with the above recommendations, the scope of the Release 3.0 initiative is defined in the following activities:

3.4.1 Reusable Common Services

The Reusable Common Services provides common technical and business services that are required for virtually every web-based application within SFA. These services can be built once and reused across all Java, web-based applications. Currently, each application is building its own set of services. These services are typically not efficient, optimally constructed, nor reusable. Additionally, these services require deep technical skills that most application teams do not have.

During the ITA R2, the first release of the RCS was rolled out and was widely accepted by application teams including FAFSA 6.0, eCB, IFAP, EIP Phase 2, Students.gov, and Policy Guidance. The ITA R3 is targeting the following services to be built: Servlet Framework, Session Framework, Scheduler, SOAP, UDDI and XML Helper.

The benefits for this initiative include:

- Substantial cost avoidance associated with building each function once and reusing for each subsequent application.
- Cost saving projected at $35,000 for each component reused in an application.
- Estimated cost saving of $280,000 per application when all eight services are used.
- Reduced hardware costs due to efficiency of well-constructed common services.
- Allow shorter time to market, shorter development cycles, and higher quality of code.
- Reduced maintenance costs associated with a single set of code rather than each application maintaining its own set of services.
- The RCS are deeply technical services that are required by most applications. By making use of these services, projects do not have make use of deeply technical and expensive resources. Cost saving in terms of staffing could be as high as $100,000 per month.

3.4.1.1 Configuration Service

Both the architecture and the applications being developed on the IBM WebSphere require configurations to be loaded from an external file. In the case of the architecture, the logging framework must load in the initial configuration of the logs. Within the application, properties containing database information must be loaded. Both the architecture and the applications utilize different approaches to reading in the files, making the code complex and non-standard. This function of the configuration file reading can be abstracted into the architecture, providing a

standard approach to the reading of all configuration files. By providing a single source from which to obtain the configurations, the impact on application developers is reduced when adding a parameter to this file.

The configuration framework will provide a standard for configuration file input. The configuration settings for an application will reside in a single implementation of the configuration manager. The mechanism for reading the configuration files should be abstracted into the architecture layer to avoid any duplicate file reading logic being written within the application code. The configuration framework will also abstract the representation of configuration files (the style in which they are written) so that if the files change formats, only the configuration framework will require coding changes – no coding changes will need to be made within the application or architecture code.

The configuration framework will load the configuration file at startup, cache the configuration into memory for easy access, and provide the functionality of searching through the configuration. It also provides the capability of determining the operating system on which it currently resides.

3.4.1.2 Web Conversation Framework

Servlets are major application subsystems. When executed, they divide into one or more conversations. Each conversation consists of a set of request/response exchanges to complete a single activity. Each conversation in turn is divided into one or more commands that represent a single request/response exchange between the client and server. Together these servlets, conversations, and commands map to classes for conversation, exchange activity, and exchange presentations that represent the core of the servlet framework. The conversation class correlates commands with exchange activities and exchange presentations. The exchange activity class executes that business logic of the user requests. The exchange presentation class formulates and sends a response to the user.

RCS Web Conversation Framework will leverage the Jakarta Struts Framework. The Struts Framework is built on the Model-View-Control (MVC) Model 2 paradigm. ITA will roll out Struts and educate application teams on the development of MVC Model 2 applications.

3.4.1.3 Scheduler Framework

Scheduler is a support tool for writing and activating batch programs based on pre-set time. The need for a scheduler becomes apparent when tasks are executed periodically and without much supervision. The framework can also be applied in the situations where a large number of repetitive tasks are performed. Users can utilize the framework to set the tasks to run at the time when system resources are the most available for efficient processing. Like other RCS frameworks, Scheduler will be built on a Java foundation.

3.4.1.4 Session Management

The session management framework simplifies, standardizes, and extends the use of session information within the J2EE standard. The session wrapper class provides a common way to access session information, decouples session information from the request, session, and application J2EE contexts, and wraps WebSphere session extension classes. The session utility class provides utilities for sessions that are common across enterprise applications, such as detecting whether a browser accepts cookies.

The session class for large sessions provides a best practice and higher performance implementation for applications requiring storage of large amounts of session information. The session framework is integrated with the servlet framework for seamless use.

The session framework defines and specifies usage of the following components:

- Session wrapper class
- Session utility class
- Session class for large sessions

3.4.1.5 JSP Tag Library

JSP tag libraries define declarative, modular functionality that can be reused by any JSP page. Tag libraries reduce the necessity to embed large amounts of Java code in JSP pages by moving the functionality provided by the tags into tag implementation classes. In doing so, tag libraries make authoring JSP pages easier and faster for the application developers.

The RCS JSP Tag Library focuses on the presentation layer of SFA applications. Application developers can use RCS JSP Tag Library to set structure and format data for web presentation. ITA will either construct the tag library of its own or leverage existing library to suit SFA's specific needs.

3.4.1.6 XML Helper

XML has gained popularity among developers in the recent years. Its application is extensive and is used in many different ways such as information exchange and configuration setting. RCS XML Helper is going to provide two key functions for the application developers. First, RCS XML Helper is going to provide XML parse function. It is able to take a XML document and extract the content. Second, RCS XML Helper is going to provide translation function between application and XML. That is, it can take configuration files or objects and translate them into XML files and vise versa.

3.4.1.7 File Transfer Protocol Service

The File Transfer Protocol (FTP) service provides the ability to transfer files between different platforms in the SFA environment. The FTP framework is an extension of a standard file transfer application which provides a Java native interface to the "put" and "get" process. Files can be moved from one server to another regardless of platform. Exception handling will be integrated with the framework to ensure that files have been accurately and completely transferred. This framework can be used for online processing within a Java thread of execution or in a batch mode where it would be called from an NT scheduled job or UNIX cron job.

3.4.1.8 Object Pooling

Object pooling is the sharing of already instantiated objects among processes. The purpose of using object pooling is to achieve higher performance. Object pooling allows processes or threads to “borrow” an object for the length of the process. Once the process expires, the object is returned to the pool. The advantage of this is cutting down the overhead on initializing and garbage collecting objects every time an object is needed. RCS Object Pooling framework will provide an architecture and API for the developers to take advantage of object pooling in their applications.

One of the most used and probably the most familiar object pooling example is database connection pooling. Database connections are initialized and pooled for processes to use. When an application requests a database connection, it establishes connection through the connection object in the pool. Once connection is no longer needed, the connection object is returned. By doing so less frequently the object initialization and garbage collection occur and hence increase performance.

3.4.1.9 Technologies Enabling Web Services

Today’s Internet consists largely of islands of web sites that cannot easily interact with each other. Web Services enables web sites to offer up functionality in a standard, programmatic fashion to other sites thus allowing the aggregation of services and interaction on a user’s behalf.

A “Web Service” is an arbitrary piece of self-contained, descriptive logic packaged in such a way that it can be located and invoked programmatically over the Internet using a set of low overhead, open standard network and application protocols (i.e. UUDI, SOAP, XML, etc.).

Web Services are, by their nature, loosely coupled services. The Web Services programming model is a service-oriented model based on the interchange of documents as opposed to arguments. The inputs, outputs, and return codes are all typically XML documents. By focusing solely on these document-based messages as interfaces, Web Services are completely language, platform, and object-model independent. Therefore, there is no reliance on an underlying framework such as .NET, J2EE, Forte, etc.

Since Web Services are an evolution of existing distributed computing models, they deliver many of the benefits of client-server and net-centric computing. Like components, Web Services

represent black-box functionality that can be reused without further analysis about how the service is implemented. Additionally, Web Services promise a few key areas of differentiation.

- Provide a straightforward, low entry cost mechanism for system-to-system interaction between companies
- Based on a set of industry standard protocols and technologies available on all platforms
- Support the reuse and extension of existing components / applications
- Enables just-in-time integration - allowing disparate business services to be federated into a composite, value-added business function without intervention from a programmer
- Enables a distributed architecture that is data / service-oriented as opposed to today's presentation-oriented model

In order to deliver the fundamental operations of Web Services, the architecture must rely on a core series of standards enabling specifications and technologies. These enablers supply the foundation for the definition of a common communication framework for Web Services. There are a number of standards bodies (i.e. W3C, OASIS, etc.) and vendors (IBM, Microsoft, etc.) collaborating to define these standards today.

3.4.1.9.1 Simple Object Access Protocol (SOAP) Framework

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols such as with HTTP and HTTP Extension Framework.

The RCS SOAP framework is an API implementation. The framework will handle the common mundane tasks such as setting XML document structure and allow application the developer to focus on building SOAP applications. The RCS SOAP framework, like other reusable common services, is built on Java foundation.

3.4.1.9.2 Universal Description, Discovery & Integration (UDDI) Framework

UDDI is a specification for distributed Web-based information registries of Web services. UDDI is also a publicly accessible set of implementations of a specification that allows businesses to register information about the Web services they offer so that other businesses can find them. The core of UDDI is its XML formatted registry. The registry describes the kind of web services an organization offers and since it is shared across the web, others can search through the registry to discover the services.

The UDDI specifications consist of XML schema for SOAP messages, and a description of the UDDI API specification. Together, these form a base information model and interaction framework that provides the ability to publish information about a broad array of Web services.

The RCS UDDI framework is an API implementation for publishing web services at SFA. The application developers can integrate the framework into their application to facilitate rapid application building.

3.4.1.10 Portlets

Portlets are reusable components that can be used in isolation or in combination with one another. Portlets are especially useful in a portal application development effort, as developers can choose which components to be available in their portal application and easily plug the needed portlet components into an application.

Each portlet in a horizontal portal is generally a separate Model-View Controller (MVC) application, loosely linked to the overall portal by ancillary requirements, such as security and visual integration (common look and feel). For productivity purposes it is desirable to partition the work so that a team of developers can work on portlets independently. Key to independent portlet development, it is imperative to have a framework available that provides a common set of MVC facilities, including:

- A navigation framework, which standardizes the mechanism used to route request processors and the target of those requests (JSP or HTML).
- JSP Tag Libraries, which simplify the development of the portlet JSPs, since they reduce the amount of JSP scripting that needs to be provided.
- A simple means of binding portlet JSP parameters to actions by means of Java Bean that has a set of properties, each corresponding to an HTML form or query parameter.

Portlet components include:

- **Calendar:** The calendar portlet displays a tabular calendar view and detail for the selected day. Selection of a new day or next/previous month with the current day displays the detail HTML links for the selected day.

- **Logon:** The Logon portlet presents a logon view, allowing the user to enter username, password, and if registered, redisplays the portal in logged-on state.

- **Registration:** The Registration portlet allows either a non-logged on user to create a new registration and logon or a currently logged on user to edit their registration information.

- **Personalization:** The Personalization portlet allows a user to edit, delete or add new personal bookmarks/links that will be displayed on the SFA links page when they are logged on.

- **Headlines:** The Headlines portlet either displays the current day's SFA headlines or allows a user to request a display of all archived SFA headlines.

- **Search:** The Search portlet allows simple search or advanced search. The latter allows the entry of search terms and selection of those sites against which the search should be performed.

  Submitting the search in either case invokes a query on the Autonomy Search engine via the RCS tag interface. All presentation and business functionality is implemented via the RCS tag interface.

- **Feedback:** The Feedback portlet allows users to provide feedback through JSP graphical interface. Users can fill out the online survey and submit it to the application. The Portlet offers two functionalities. It takes user input and sends out an email containing the feedback to a designated email address. The portlet also provides a database storing capability. It can take user inputs and store them into a database table. Depends on the success and failure of each procedure, the user will be directed to appropriate JSP page at the end of the feedback process.

3.4.2 ITA Core Services

This effort will provide for the integration of ITA maintenance and new ITA development; this structure is known as a high performance team. Services that have been rolled out in previous releases will receive bug fixes, modifications and enhancements. Additionally, this effort will provide operational support for the ITA environment and product suite for applications that are in production. This effort will also provide the ITA the ability to roll in the RCS to applications that have already moved into production. Tasks include:

- Bug fixes and enhancements to Reusable Common Services from ITA R2

- Technical Architecture troubleshooting for Best Practices from ITA R2

- Ensure ITA products are running in production, includes monitoring, restarts & troubleshooting

- Upgrades, which ensure the latest product enhancements and functional improvements are available to customers increasing end-user satisfaction.

3.4.3 ITA Software

In order for the ITA to execute the services listed above, software must be procured. The following software is needed to meet the requirements of this task order:

- Performance Tools: Jprobe and LoadRunner

- XML Integrated Development Environment: XML Spy,

- Testing Tools: Junit

3.4.4 Performance Enhancement Services

In order to maximize server utilization efficiency and system response time, a set of tools is needed to analyze and monitor application performance in development, test and production. A set of performance tools will allow us to plan for current and future capacity, handle peak volumes, and staff the appropriate level of call center support resources. In addition to performance tools, this task will also provide a set of performance testing best practices and sample performance test deliverables from other Accenture projects. The benefits for this initiative are:

- With the use of monitoring, tuning and performance testing tools, SFA applications can be properly sized and hardware can be appropriately assigned rather than using a historical solution of adding more hardware.

- Performance Enhancement Services will allow applications to identify and solve performance bottlenecks before the application goes into production.

3.4.5 SFA Technical Architecture Practices

A set of best practices that addresses how to design efficient web based applications with a similar look and feel as well as similar technical standards. Application teams do not need to do research in deeply technical areas as the research is done once and rolled out to each application team. The Best Practices for ITA R3 include Enterprise Java Bean usage, as well as guides on SOAP, UDDI, performance testing, and application load balancing.

3.4.6 ITA Scripting

Automates the manual process of repetitive technical tasks within the ITA. Scripting the repetitive ITA tasks allows for an increase in team efficiency and reduces setup time for new application teams. Once the ITA creates scripts for repetitive tasks, they no longer have to allocate a developer to build environments, or set up servers. These tasks include:

- Building development, test, performance test, and production environments

- Building application and web servers

- Building automated startup and shutdown scripts

3.4.7 ITA Workshops

As a major component of the ITA communication strategy, a series of bi-monthly workshops are scheduled over the task order's lifecycle. The purpose of the workshops is to raise the awareness of application teams with respect to the services provided by the ITA. The workshops also provide a forum for discussion for respective audiences with whom the ITA interacts. The topics that the ITA workshops will cover are: ITA Overview, Reusable Common Services, ITA

Performance Testing, FAFSA Architecture, and Fundamentals of High-Availability Applications. The first ITA workshop will take place on January 30, 2002.

3.4.8 ITA Product Specialty Support

Products deployed in SFA are essential to each application. It is the greatest importance for the products used in each application to function at an optimum level. With research and knowledge acquisition, ITA has developed deep expertise in the products it recommended. ITA has been and will continue to support products such as IBM HTTP web server and WebSphere Application server.

3.4.9 Independent Verification & Validation

In order to maximize the performance of the ITA services, independent verification is needed to pinpoint problems and show areas of improvement. This will provide metrics and performance data to improve the systems and components within ITA Release Three. Tasks include:

- Performance services and tests to validate technical architecture and aid in accurate capacity planning.
- Independent Verification
- Common Services

4 SFA Projects

4.1 Overview

Based on existing documentation and interviews with application leads, this section of the strategic assessment documents the following information for each SFA project:

Section	Description
Functional Description	High level overview of the application's functionality
Technical Description	Technical description of the application including volume of users, concurrent users, components (such as HTML and JSP pages), and products
Status	Describes the current status of the application
ITA 2.0 services used	Identifies the ITA 2.0 services used by the application
Enhancements / Upgrades	Enhancements and / or upgrades required by the application
Current / Potential ITA 3.0 services	Lists current ITA services and potential ITA 3.0 services that may be used by the application in its next release
Schedule	Schedule of the application's development, testing, and production dates

4.2 eCampus Based (eCB)

4.2.1 Functional Description

The eCampus-Based (eCB) enables the Department of Education (ED) to provide more than $2 billion in Title IV student financial assistance funds to approximately 4000 post-secondary institutions each year through a complex allocation model. The system provides allocations / authorizations for grant, work-study, and loan funds to these institutions, and the institutions, in turn, use these funds to provide student financial assistance to more than 1,000,000 students each year. There are several distinct programs, each with its own legislative history and regulations, within the Institution-Based Programs. These include the Federal Supplemental Educational Opportunity Grant Program (FSEOG), Federal Work-Study Program (FWS), and Federal Perkins Loan Program (Perkins).

The eCB application provides the basis for funding / allocations and reallocation of funds to schools for needy students, the Fiscal Operations Report which summarizes the Institution-Based activity for a specific award year, and the Application to Participate process, which enables

schools to request a continuance and/or an increase in their participation under the Institution-Based programs.

In summary, the eCB application calculates funding formulas, provides the basis for funding of schools for FSEOG, FWS, and Federal Perkins Loan programs. In addition, it supports the filing of the Fiscal Operations Report and Application to Participate (FISAP) via data received through EDConnect/SAIG.

As part of the overall Modernization program, the SFA sought to redesign the Institution-Based Programs' system to provide Institutions and staff with an integrated and intuitive solution that increases information access and self-service capabilities, therefore, supporting SFA's goals of reduced unit cost, customer satisfaction and employee satisfaction.

ITA 2.0 Products
• Visual Age for Java Enterprise 3.5 • WebSphere Studio Advanced Edition 3.5.3 • Sun Solaris Version 2.6 • WebSphere Application Server 3.5.3 • Oracle 8i RDBMS Version 8.1.7 • IBM eNetwork Dispatcher

4.2.2 Status

eCB is currently in development. It is scheduled to be in production in February 2002.

4.2.3 ITA 2.0 Services Used

The following are the services that eCB used in ITA R2.0:

ITA Services	Service Components	Services Used
Environment Support	? Set up development, test, performance test, and production environments ? Troubleshooting	v
Product Support	? WebSphere ? IBM HTTP Server ? Visual Age for Java 3.5	v
Reusable Common Services (RCS)	? E-Mail Framework ? Logging Framework	v
SME	? Design and Code review ? ITA coding standards ? Performance Testing	v

4.2.4 Enhancements/Upgrades

eCB Fix Release 3 has been proposed to patch up several eCB release 1 production issues.

4.2.5 Potential ITA 3.0 Services

The following is a list of potential ITA 3.0 services that can be used by eCB 2:

Potential ITA Services	Service Components
Environment Support	
Product Support	? WebSphere ? IBM HTTP Server
Reusable Common Services	? SOAP Framework ? UDDI Framework

	? Configuration Service ? Servlet Framework ? Scheduler ? Session Framework ? JSP Tag Library ? XML Helper ? FTP ? Object Pooling Framework ? Content Label
SME	Performance Testing

eCB 2 is in the process of implementing RCS Email framework for its mass emailing function. eCB 2 gave positive feedback on ITA RCS. eCB is looking forward to the next release of RCS. eCB also praised on the strength of ITA's ability to provide environment building and maintenance services.

4.2.6 Schedule

eCB 2 is currently doing system test.

Task	Start Date	End Date
Development	11/01/2001	12/12/2001
Test	12/12/2001	02/15/2002
Performance Test	02/18/2002	02/25/2002
Production		Mid March

4.3 Common Origination and Disbursement (COD)

4.3.1 Functional Description

The Common Origination and Disbursement (COD) process enables SFA to deliver financial aid for Pell Grants, Direct Loans, and campus-based programs through a single process. A common process and system to support origination and disbursement will be critical if SFA hopes to

achieve an enterprise-wide solution that will provide real-time data to students, schools, and financial partners via web portals. COD is a transaction-based payment method for schools. The team designed a "baseline" process that will be available for all schools. They also identified some areas in the process that could be used as "performance-type" indicators, thereby providing additional options and flexibility to schools that meet these measures.

- WAS 3.5.3
- IHS 1.3.12.2
- MQSeries 5.2
- MQIntegrator 2.01

4.3.2 Status

COD is currently in the testing stage.

4.3.3 ITA 2.0 Services Used

COD used the following ITA 2.0 services:

ITA Services	Service Components	Services Used
Environment Support	? Development environment	v
Product Support	? WebSphere ? IBM HTTP Server	v
Reusable Common Services (RCS)	? Exception Handling Framework ? Logging Framework	v
SME		

4.3.4 Enhancements/Upgrades

COD is currently in release 1. There is no previous release to compare.

4.3.5 Schedule

Task	Start Date	End Date
Release 1.0a		9/25/01
Release 1.0a Test	9/25/01	11/16/01
Release 1.0b		10/23/01
Release 1.0b Test	10/23/01	1/08/02
Release 1.0c		11/20/01
Release 1.0c Test	11/20/01	1/20/02
Integration/Performance Test	1/14/01	3/08/02
Release 1.0 in Production		3/18/02
Release 1.1 in Production		4/29/02

4.4 Enterprise Architecture Integration (EAI)

4.4.1 Functional Description

The Enterprise Application Integration (EAI) project is an ITA R2.0 effort that will create an SFA EAI architecture for connecting front end systems to back end legacy systems via a reusable common architecture. EAI is a set of technology services that enables the sharing of processes and data of disparate systems to support end-to-end business processes. The EAI architecture enables the many "stovepipe" applications to exchange information via common, reusable methods and infrastructure. This application will provide the core messaging and sample transformation functionality to validate the EAI architecture and connectivity between the WebSphere Server and release 1.0 legacy systems.

The EAI will allow the SFA to integrate new web-based applications with existing back-end systems, while at the same time, providing a means to migrate away from reliance on existing legacy systems.

EAI provides capabilities that will allow for the integration of web-based applications, the Data Warehouse environment, commercial-off-the-shelf (COTS) packages, and existing legacy systems within the SFA technical environment.

The EAI architecture provides the following technical services:

- Communications Middleware
- Transformation and Formatting
- Application Connectivity
- Business Process Management

4.4.2 Status

The EAI is currently in the testing phase. The team has finished its Loan Origination System performance test on November 9, 2001.

4.4.3 ITA 2.0 Services Used

EAI used the following ITA 2.0 services:

ITA Services	Service Components	Services Used
Environment Support	? Development, Test and Production environments	v
Reusable Common Services (RCS)	? Exception Handling Framework ? Logging Framework	v

	? Persistence Framework	
SME		

4.4.4 Enhancements/Upgrades

No enhancements or upgrades for EAI Release 2. Assessment meetings are currently held to determine enhancements or upgrades for Release 3.

4.4.5 Schedule

Time frame is based on all EAI adaptors.

EAI 2.0		
Task	**Start Date**	**End Date**
Development/Test	6/01/01	1/07/02
Production	11/16/01	1/14/02
EAI 3.0		
Task	**Start Date**	**End Date**
Analysis	11/01/01	1/31/02
Development	TBD	TBD
Test	TBD	TBD
Production	TBD	TBD

4.5 Enterprise Information Portal (EIP)

4.5.1 Functional Description

The Enterprise Information Portal (EIP) is a web-based enterprise wide portal, with separate views based on the audience. The application will consist of five separate views catering to the audience, namely Students, Schools, Financial Partners, Employees of SFA, and a public view for the general Internet audience. These views will contain information pertaining to their respective channel, and will allow for a seamless and transparent distribution of pertinent information from a variety of back-end data sources, and assist the users in their daily work.

The EIP effort is a replatforming of the existing Schools Portal application.

4.5.2 ITA 2.0 Services Used

EIP used the following ITA 2.0 services:

ITA Services	Service Components	Services Used
Environment Support	? Set up development environment	v
Product Support	? WebSphere ? IBM HTTP Server	v
Reusable Common Services (RCS)	? Exception Handling Framework ? Logging Framework ? Persistence Framework ? Search Framework	v
SME		

4.5.3 Enhancements/Upgrades

The current EIP is the first release and not future releases are planned.

4.5.4 Potential ITA 3.0 Services

EIP can potentially use the following ITA 3.0 services:

Potential ITA Services	Service Components
Environment Support	
Product Support	? WebSphere ? IBM HTTP Server
Reusable Common Services	? SOAP Framework ? UDDI Framework ? Configuration Service

	? Servlet Framework ? Scheduler ? Session Framework ? JSP Tag Library ? XML Helper ? FTP ? Object Pooling Framework ? Content Label

4.5.5 Schedule

Task	Start Date	End Date
Analysis	4/06/01	4/20/01
Build	4/20/01	5/01/01
System Test	5/01/01	6/15/01
Performance Test	6/15/01	7/15/01
Production/Go Live	7/31/01	7/31/01

4.6 Free Application for Federal Student Aid (FAFSA)

4.6.1 Functional Description

Free Application for Federal Student Aid (FAFSA) on the Web is an application used by college students and schools to submit financial applications via the Internet. During the 2000-2001 academic year, over two million students will use FAFSA on the Web to apply for Student Financial Aid. SFA expects the volume to double during the 2001-2002 academic year and to continue to rise in future years.

The application replaced a paper form and previous version located at http://fafsa.ed.gov. The online application includes approximately 80 questions that are used to determine students' eligibility for financial assistance. Students can create, review, renew, or correct forms by accessing the site with a Personal Identification Number (PIN).

The FAFSA on the Web modernization initiative is an effort to modernize FAFSA on the Web while accommodating the anticipated increase in users. The initiative will improve customer satisfaction by providing students who use the site a complete user experience in applying for Federal, State, and Institutional aid. Customer satisfaction will be improved by increasing the application's usability, accuracy, efficiency, security, and supportability/scalability. SFA will also benefit from the reduction of costs due to reduced manual processing of applications and fewer customer service calls as a direct result of an application that is easier to use.

Usability will include all day everyday access, a real-time display of results from the Central Processing System (CPS), reusable data that can be shared between different applications, and the incorporation of the latest best practices in web user interface design.

4.6.2 Status

FAFSA release 6.0 has gone live and FAFSA 7.0 is in the planning stage.

4.6.3 ITA 2.0 Services Used

ITA R2.0 has provide FAFSA with the following support:

ITA Services	Service Components	Services Used
Environment Support	? Build and configure development environments ? Build and configure test environment ? Build and configure production environment ? Participated in all sixteen performance tests	v
Product Support	? WebSphere ? IBM HTTP Server ? Network Dispatcher ? Provide WebTrends reports	v
Reusable Common Services (RCS)	? Logging Framework	v
SME	? Java coding standards ? Code and Design Review	v

	? Capacity planning ? Performance testing	

4.6.4 Enhancements/Upgrades

FAFSA 6.0 extends FAFSA 5.x and transforms architecturally and technologically. Architecturally, FAFSA 6.0 rebuilds FAFSA 5.x servlets in Java 2 Enterprise Edition (J2EE) framework as well as the inclusion of JSP's. Technologically, FAFSA 6.0 uses IBM WebSphere Application Server for its web applications.

4.6.5 Potential ITA 3.0 Services

Future FAFSA releases may be able use the following common services:

Potential ITA Services	Service Components
Environment Support	
Product Support	? WebSphere ? IBM HTTP Server
Reusable Common Services	? SOAP Framework ? UDDI Framework ? Configuration Service ? Servlet Framework ? Scheduler ? Session Framework ? JSP Tag Library ? XML Helper ? FTP ? Object Pooling Framework ? Content Label

4.6.6 Schedule

Task	Start Date	End Date
FAFSA 6.0		
Test	9/11/01	12/01/01
Performance Test	9/24/01	12/31/01
Production		1/01/02

4.7 Financial Partners (FP) Data Mart

4.7.1 Functional Description

The Financial Partner's (FP) Data Mart provides executive information and decision support capabilities around several key business functions for both the Guaranty Agencies (GAs) and the Lenders. The scope for the Data Mart initiative establishes the Data Mart infrastructure within the Channel and provides initial Risk Management, Customer Relationship Management, Compliance Management, and Portfolio Management functionality related to the thirty-six Guaranty Agencies and the approximately four thousand lenders.

The FP Channel is responsible for both current and retired loan programs and, as such, is under customer and program obligation to ensure the capture and comparative ability of all programs. By collecting information (associated with student lending among Financial Partners including Guaranty Agencies, Lenders, and Servicers particularly in the FFEL Student Loan Program) from several sources into a central location, personnel in the Channel as well as external partners will be able to more efficiently identify areas in which each party may assist the other while improving the support for students within the Federal Family Education Loan Program.

The FP data mart receives data from the National Student Loan Data System (NSLDS), The Federal Family Education Loan System (FFEL), The Post-secondary Education Participants System (PEPS), and FMS. Using this information, the FP data mart performs reporting and analysis on loan level data and Individual Lender and Guarantee Agencies. The FP data mart will include Analysis, Decision Making, Trend, and Risk Assessment Requirements reporting. Various reporting (i.e., the relationships between Guaranty Agencies, Lenders, and Servicers; identification of 'top' Guaranty Agencies and Lenders; the risk scorecard) will be developed to support some of the analytic and decision support processing that is required. Analytic and decision support reporting includes 'what-if' analysis, trending, and benchmarking in areas related to the operations, default management, loan servicing, financial information, and regulatory requirements of the Guaranty Agencies and Lenders.

The Data Mart is designed to be implemented in three releases in order to populate the Data Mart with information from the identified four source systems. Release 1 established the Data Mart infrastructure within the Channel and populated the Data Mart with Guaranty Agency and Lender information from the FFEL system. Release 2 will populate the Data Mart with loan level information from NSLDS. Release 3 will populate the Data Mart with current Guaranty Agency and Lender reports from FMS and with information related to performance reviews and audits from PEPS.

4.7.2 Status

FP Data Mart release 1 is currently in production.

4.7.3 ITA 2.0 Services Used

ITA 2.0 set up the IBM Network Dispatcher for FP Datamart. The ITA team configured Network Dispatcher for FP Data Mart.

4.7.4 Enhancements/Upgrades

FP Data Mart is using the current product installations. The FP Data Mart is planning to upgrade to MicroStrategy 7.1 Web & Intelligent Server and Oracle 8.1.6.

Release 2 of FP Datamart is currently in the requirements gathering phase.

4.7.5 Potential ITA 3.0 Services

4.7.6 Schedule

Release 2 of FP Datamart is in requirements gathering phase.

4.8 Information for Financial Aid Professionals (IFAP)

4.8.1 Functional Description

The Information for Financial Aid Professionals (IFAP) web site (http://ifap.ed.gov) is an electronic library for financial aid professionals containing publications, regulations, and guidance regarding the administration of the Title IV Federal Student Financial Aid Programs. IFAP is managed by the Customer Support Branch (CSB) within the Office of Student Financial Assistance Programs (OSFAP) of the Department of Education.

This version of the IFAP web collection went “live” in early March 2001. IFAP consists of forty-one (41) specific types of publications that CSB posts.

IFAP has a subscription option that notifies registered customers via email when new information (Dear Partner Letter, Announcement, Federal Register, etc.) has been added to the wealth of information available on IFAP. The email is customized based on users’ subscription selections.

Users can select to be notified when new documents are added to the IFAP catalog based on Publication Type and Program / Service categories. Users can select more than one Publication Type or Program / Service category. Even if selections overlap, the system is designed to not send duplicate notifications of a single document.

The IFAP subscription option sends "Weekly" (a weekly summary of all IFAP additions) subscription emails and "48 Hour" (a summary of IFAP additions within 48 hours) emails based on the end user's subscription options form. Users must be registered to use the IFAP subscription option.

4.8.2 Status

IFAP is in production. There are no future releases planned. Bug fixes and enhancements are ongoing.

4.8.3 ITA 2.0 Services Used

IFAP used the following services from ITA 2.0:

ITA Services	Service Components	Services Used
Environment Support	? Set up development, test, and production environments ? Maintain development, test, and production environments	v
Product Support	? WebSphere ? IBM HTTP Server ? Interwoven TeamSite ? Autonomy	v
Reusable Common Services (RCS)	? Search	v
SME		

4.8.4 Potential ITA 3.0 Services

IFAP is expected to utilize ITA 3.0's environment support and SME in operations.

4.8.5 Schedule

IFAP is currently in production. No new versions are planned within ITA R3.0.

4.9 Operations

4.9.1 Functional Description

The Operations team is responsible for maintaining applications that are currently in production. Applications such as IFAP, Schools Portal, Financial Partner Datamart (FP Datamart), and Chief Financial Officer Datamart (CFO Datamart) are under the care of Operations team. The Operations team is also the front-end to client and is in charge of communicating the needs of the client to the Modernization Partner Teams. Operations team's tasks also include performing simple change requests for applications, and maintain application availabilities on the application level and infrastructure level. The infrastructure support, or the Technical Architecture support, is being transferred to ITA 3.0. After the transfer, Operations team will only focus on the applications maintenance.

4.9.2 Status

Maintenance of applications is ongoing. Changes to applications are sent from the clients through Simple Investigation Requests (SIRs). Infrastructure support has been transferred to ITA 3.0.

4.9.3 ITA 2.0 Services Used

The Operations team used the following ITA 2.0 services:

ITA Services	Service Components	Services Used
Environment Support		
Product Support	? WebSphere ? IBM HTTP Server ? Autonomy ? Interwoven	v
Reusable Common Services (RCS)	? Search	v
SME	? Unix Scripting ? Trouble-shooting production application	v

4.9.4 Potential ITA 3.0 Services

The Operations team can use the following ITA 3.0 Services:

Potential ITA Services	Service Components
Environment Support	Upgrades
Product Support	? Interwoven
Reusable Common Services	
SME	? Unix Scripting ? Trouble-shooting out of production application

5 Appendices

Appendix I – Technical Description for SFA Projects Technical descriptions for each of the SFA application teams to complement the functional description above.	"Technical Descriptions.doc"
Appendix II - Application Services Matrix Depicts a summary view of the ITA services and the application teams that have benefited from them.	"ITA Services Matrix.ppt"
Appendix III – Strategic Assessment Questionnaire The questionnaire employed by the ITA team in interviews with the application teams.	"Strategic Assment Questionnaire.doc"