

Application Model for ASG

Methodology

The ASG application will utilize a Model-View-Controller architecture within the Struts Framework.

Framework

This application will leverage the Struts framework, utilizing Servlet controllers to connect to EJB business-logic resources and deliver end-user JSP views.

Each module within the application will be a self-contained WAR. All WARs will be packaged into an EAR for installation and distribution.

Component Categories

The ASG application model is divided into the following component categories

- EJBs—these will be used as the fundamental method for carrying out activity between the application and the datasource. They will be called from the servlet controllers. Module functionality of multiple beans will be packaged as JARs.
- Servlets—these will provide controlling actions for module functionality and connect client requests to EJBs
- Page components—these are JSP pages representing client output and providing for user input

All of these components will be collected into one or more WARs that correspond to the master modules within the application (parenthetical annotations represent use case designations)

- Order Handling (OH1)
- Portfolio Reports (RE1)
- Portfolio Analysis (AN1)
- Account Management (AC1)
- Administration Module (AM1)
- SubAccounting

In addition, there will be a number of system-level EJB JARs developed that will be accessible from all WARs and packaged into the EAR

- Database Communication (DC1)
- Transaction Processing (TP1)
- Outside Feed Processing (OF1)
- Credential Processing (CP1)
- Database Processing (DP1)
- Data Validation (DV1)
- Data Calculation (DA1)
- Application Errors (AE1)
- Graphing (GR1)
- Error Processing (EP1)

Java APIs

This application will utilize the following J2EE 1.3 APIs

- Java Enterprise Message—this will serve to communicate between this application and other applications in which it may be integrated.
- Java Transaction—this will serve to handle atomic transactions (i.e., such as updating account balances based upon sales and orders)

Data Manipulation

Data manipulation will be handled in two ways

- Reading and Displaying—this interaction with data will come from data stored in session or memory
- Writing—this interaction with data will happen via a session-based connection to the datasource

Data Validation

Data validation happens in three levels

- Field—each field within a form is validated for proper entry
- Form—the entire data set from a complete form is validated against proper entry
- Account—account information (for buying and selling securities) is validated

Usage Hierarchy

The following is a usage hierarchy and nomenclature for stored information

- User—the person who logs into the system and access data and functionality
- Client—customers of the user
- Account—a representation of the client's cash and securities balances
- Portfolio—a representation of the client's current securities assets

Issues

The following table describes issues, document version in which issue was included, response(s), and response(s) date(s).

Issue	Document Version	Response(s)	Response(s) Date(s)
Should form validation be at the master controller level or individual controller levels?	Draft 2.2	Individual controller level	6/7/2003
Should we use the ANT specification for packaging modules, rather than WARs?	Draft 3		

Application Server

The application server that will be used to implement this application is Orion.

Component Library

Each of the following tables is a breakdown, by module, of the components. There will be a controller servlet for each module

Master Controller

There will be a master Servlet controller that will direct client requests to module controllers through individual servlet mappings.

Name	Type	Component Relationships and Dependencies	Component	Type	Description
ApplicationController	Servlet	All module controllers	See below		
			GetClients	Action Class	Controller class that is part of Struts 2.0 Framework (org.apache.struts.actions.DispatchAction); sends specific types of requests from this module to the appropriate module servlet
			GetClientData	Action Class	Get client data based upon available clients returned from GetClients; hold data in session
			Login	Action Form	Gather requested login information from the client form and pass to the Credential Processing Component (CP1)
			ErrorCapture	AppException Class	Capture the error and send to the error processing component for storage in the DB
			CredentialCheck	Action Class	Check session information for credentials; redirect where appropriate and necessary
			ErrorMessage	Action Message Class	Show translated error message to end user

Order Handling (OH1)

Name	Type	Component	Component	Type	Description
------	------	-----------	-----------	------	-------------

		Relationships and Dependencies			
OrderHandling	Servlet	All objects below	See below		The controller servlet for this module; This servlet controller utilizes the Struts 2.0 class library to communicate with EJBs and other application components. Componnets listed underneath this controller are called by controller Execute methods.
			ValidateDataField	Action Class	Validate individual field entries by utilizing an in-session, instantiated bean from the DataValidation system component
			OrderForm	ActionForm	Call appropriate form for editing based upon 1) buy or sell and 2) type of asset; form types are defined in XML; need to instantiate the appropriate component from the Data Validation component for the field-level validation that will occur with this form.
			OrderSubmit	Dispatch Class	Get data from form and send form data onto the appropriate system components (validation) and return results (i.e., success or failure from database communication and/or validation) along with the data. If success, send to PlaceOrder and display appropriate message in ValidationAlert; if bulk order, send appropriate tag
			PlaceOrder	Action Class	Send the validated information onto the Transaction processing component; if bulk order, send appropriate tag for looping request.
			ValidationAlert	Action Message Class	Send any messages about validation or order placement to the appropriate area of the .jsp page
			CreatePortfolio	Action Forward	If a Buy Order placement does not have a portfolio (i.e., the user has selected the "new portfolio" from the portfolio drop-

					down when placing this order), then the user must create a new portfolio. The actual portfolio is created after the order has been processed. This sends the user to the Account Management module to create a new portfolio and/or new customer account. Because of a passed flag, the Account Management module then sends the user back to the BuyOrder page, populates form with a populate() method and the user can finish. But, the actual portfolio information is held in session until the order is processed.
			SetSubAccount	Action Class	If the user selects that this is a sub-account on the buy page, then they must select a sub-account from the drop-down menu that is populated when they select the "subaccount" checkbox; this treats the buy process as a bulk order
OrderEntry	EJB-JAR	Data communication Transaction process Order validation Error processing	See below		This processes the appropriate type of order once the controller servlet has captured the information
			GetPricing	SessionBean	Get formatted pricing from CF1. This needs to be stateless
			SubmitOrder	FormBean	Submit the order to OrderSubmit in the controller servlet
			PutOrder	SessonBean	If order is a sell order, put asset into appropriate form built by OrderForm
OrderEdit ¹	EJB-JAR	Data communication Transaction process	See below		This processes the order changes once the controller servlet has captured the information. Actual editing of the order information is carried out by a linking with

¹ This is only applicable for Buy Orders

		Order validation Error processing			the OrderEntry class. This may take place through a strut Action or ActionForm.
			GetOrderData	Entity Bean	Retrieve order information from database and populate into the right type of form
			GetPricing	Session Bean	Get formatted pricing from CF1. These needs to be stateless
ViewOrder	EJB- JAR	Data communication OrderEdit	See below		This gathers information from the database using a system component
			GerOrders	Entity Bean	Get a list of all the orders from the session or memory data using the data component
			GetOrderDetails	Entity Bean	Retrieve order information from client data in session (GetClientData); includes all order data files (i.e., status)
BulkOrder	EJB JAR	Data communication OrderHandling	See below		
			GetPricing	SessionBean	Get formatted pricing from CF1. This needs to be stateless
			OrderCapture	Entity Bean	Captures order information if the "batch order selection box" is checked on the OrderBuy.jsp or OrderSell.jsp page; places order information into session
			SelectPortfolios	Entity Bean	Select the specific portfolios (either through a multi-selection drop-down or checkboxes) upon which to apply the bulk order (either sell or buy)
			SubmitOrder	Form Bean	Create an expanded set of data (with an order request for each portfolio with each asset) and send it to OrderSubmit in the servlet with a tag that this is a bulk order; this tag will call a looping method within the validation component to validate each individual order within the bulk order

Portfolio Reporting (RE1)

Name	Type	Component Relationships and Dependencies	Component	Type	Description
PortfolioReporting	Servlet	All objects below	See below		The controller servlet for this module; This servlet controller utilizes the Struts 2.0 class library to communicate with EJBs and other application components. Componnets listed underneath this controller are called by controller Execute methods.
			GenerateReport	Dispatch Class	Call the system component to get the appropriate data and populate the screen; needs to include some sort of get method to call the appropriate report-type component from the SeeReport Jar
			ExportType	Action Class	Determine type of export file that is to be created; gather specifications for export from XML; pass to ExportData
			ExportData	Dispatch Class	Send export type parameters to the SaveData bean in the ExportData JAR to actual saving of the file
			ValidationAlert	Action Message Class	Send any messages to the appropriate area of the .jsp page
SeeReport	EJB JAR	All objects below	See below		
			OrderGeneration	Entity Bean	Instantiate database communication component with appropriate parameters for type of data to return; report data requirements are defined in XML
			PositionUpdates	Entity Bean	Instantiate database communication component with appropriate parameters for type of data to return; report data requirements are defined in XML
			PositionMaintenance	Entity Bean	Instantiate database communication

					component with appropriate parameters for type of data to return; report data requirements are defined in XML
			Inquiry	Entity Bean	Instantiate database communication component with appropriate parameters for type of data to return; report data requirements are defined in XML
			Monitor	Entity Bean	Instantiate database communication component with appropriate parameters for type of data to return; report data requirements are defined in XML
			RiskManagement	Entity Bean	Instantiate database communication component with appropriate parameters for type of data to return; report data requirements are defined in XML
ExportReport	EJB JAR				
			GetData	Session Bean	Identify data to be exported from data held within client session
			SaveData	Entity Bean	Save data in appropriate format as indicated by user selection in controller servlet

Portfolio Analysis (AN1)

Name	Type	Component Relationships and Dependencies	Component	Type	Description
PortfolioAnalysis	Servlet	All objects below	See below		The controller servlet for this module; This servlet controller utilizes the Struts 2.0 class library to communicate with EJBs and other application components. Components listed underneath this controller are called by controller Execute methods.
			CalculatePerformance	Action Class	Send parameters of specific type of

					performance request to PortfolioPerformance >> PortfolioData
			GetPortfolioData	Action Class	Instantiate database communication component to get portfolio data and put it into session bean (PortfolioData)
			ExportType	ActionClass	Determine type of export file that is to be created; gather specifications for export from XML; pass to ExportData
			ExportData	Dispatch Class	Send export type parameters to the SaveData bean in the ExportData JAR to actual saving of the file
			ValidationAlert	Action Message Class	Send any messages to the appropriate area of the .jsp page
PortfolioPerformance	EJB JAR	Database Communication Calculation	See below		
			PortfolioData	Session Bean	Capture specific data from user session based upon CalculatePerformance response; send data to Format Analysis
			FormatAnalysis	Entity Bean	Send data to calculation component for processing (specific data, not all data); create analysis container; present container to client
ExportData	EJB JAR	Database Communication Calculation	See Below		
			GetData	Session Bean	Identify data to be exported from data held within client session; data type requirements are outlined in XML
			SaveData	Entity Bean	Save data in appropriate format as indicated by user selection in controller servlet

Account Management (AC1)

Name	Type	Component	Component	Type	Description
------	------	-----------	-----------	------	-------------

		Relationships and Dependencies			
AccountManagement	Servlet	All objects below	See below		The controller servlet for this module; This servlet controller utilizes the Struts 2.0 class library to communicate with EJBs and other application components. Componnets listed underneath this controller are called by controller Execute methods.
			GetPortfolioList	Action Class	Get a list of all portfolios and populate navigation; pass specific portfolio identifier data to ShowPortfolio
			GetClientList	Action Class	Get a list of all clients and populate navigation; pass specific client identifier data to Show Client
			ShowClient	Action Class	Gather data and pass to ViewClient; client data is based upon GetClientList
			ShowPortfolio	Action Class	Gather data and pass to ViewPortfolio; portfolio data is based upon GetPortfolioList
			SubmitForm	DispatchClass	Gather client/portfolio data from form bean and submit to data validation component; return data (with message) to ValidationAlert. If message is not an error, returned data is passed to the ProcessClient component
			SavePortfolio	Action Form	Gather validated data from ValidatedData and send to Database Processing component; this process does not actually occur until have the transaction has been processed
			ValidateDataField	Action Class	Validate individual field entries by utilizing an in-session, instantiated bean from the DataValidation system component
			ValidationAlert	Action Message Class	Send any messages about validation or order placement to the appropriate area of the .jsp page

			SaveClient	Action Class	Gather returned data from SubmitClient and sent to Database Processing component
Portfolio	EJB JAR	Database Communication File System	See below		
			CreateClient	Form Bean	Create new client
			EditClient	Form Bean	Edit client information
			ViewClient	Session Bean	Display information about a client including portfolios, cash assets, and securities assets
			View Portfolio	Session Bean	Display contents of portfolio for client
			CreatePortfolio	Entity Bean	Create a new portfolio and send the appropriate data to the system components for storage in the database; THIS COMPONENT IS ONLY CALLED FROM THE ORDER HANDLING MODULE AND ONLY AFTER A TRANSACTION HAS BEEN COMPLETED

SubAccounting

Name	Type	Component Relationships and Dependencies	Component	Type	Description
SubAccounting	Servlet	All objects below	See below		The controller servlet for this module; This servlet controller utilizes the Struts 2.0 class library to communicate with EJBs and other application components. Componnets listed underneath this controller are called by controller Execute methods.
			GetSubAccountList	Action Class	Get a list of all subaccounts and populate navigation; pass specific portfolio identifier data to ShowPortfolio

			ShowSubAccount	Action Class	Gather data and pass to ViewPortfolio; portfolio data is based upon GetPortfolioList
			SubmitForm	DispatchClass	Gather subaccount data from form bean and submit to data validation component; return data (with message) to ValidationAlert. If message is not an error, returned data is passed to the ProcessClient component
			SaveSubAccount	Action Form	Gather validated data from ValidateData and send to Database Processing component
			ValidateDataField	Action Class	Validate individual field entries by utilizing an in-session, instantiated bean from the DataValidation system component
			ValidationAlert	Action Message Class	Send any messages about validation or order placement to the appropriate area of the .jsp page
SubAccount	EJB JAR	Database Communication File System	See below		
			CreateSubAccount	Form Bean	Create new subaccount by identifying available clients and adding them to the account; define what subaccount will be buying and set parameters for recurring purchases
			EditSubAccount	Form Bean	Edit subaccount information including participating portfolios/clients as well as distribution/recurrence criteria
			ViewSubAccount	Session Bean	Display contents of subaccount

Administration Module (AM1)

Name	Type	Component Relationships and Dependencies	Component	Type	Description
------	------	--	-----------	------	-------------

AccountManagement	Servlet	All objects below	See below		The controller servlet for this module; This servlet controller utilizes the Struts 2.0 class library to communicate with EJBs and other application components. Componnets listed underneath this controller are called by controller Execute methods.
			GetUsers	Action Class	Get a complete list of users upon initiation of the module; put into session or memory until user navigates to User.jsp
			ViewUser	Action Forward	Call the User package passing along the appropriate user information as attached to the navigational item
			DeleteUser	Action Forward	Call the User package passing along the appropriate user information as attached to the navigational item
			AddUser	Dispatch Class	Returning with validation information from ActionForward in ValidateData of master servlet, call appropriate component (User.AddUser); this class also covers modifying existing user data1
			SubmitUser	ActionForward	Send the submitted form information for validation (to ValidateData)
			ValidateData	Dispatch Class (with ActionForward to master controller ValidateData)	Get data from client form and pass to master controller for validation (which, in turn, passes to appropriate system component through parameterized ActionForward)
			ImportData	Action Class	Call ImportData.GetData and ActionForward to the upload data page
			ImportMessage	Action Message Class	Send user the appropriate success or failure message for importing data; identify exactly what part of data caused error

			GetUser	Session Bean	Get the user information based upon credentials passed from the servlet for editing a user; call Users.EditUser to populate form; ActionForward to user form page
			ExportType	ActionClass	Determine type of export file that is to be created; gather specifications for export from XML; pass to ExportData
			ExportData	Dispatch Class	Send export type parameters to the SaveData bean in the ExportData JAR to actual saving of the file
			ViewLog	ActionClass	Determine type of log file to return, pass requirements to Logs.GetLog; log parameters are defined by XML
			SubmitVariables	ActionClass	Call ValidateData (of master servlet) to validate the form-field entries; Action Forward will return request to the SetEnvironment.ModifyVariables component to update the system settings; should there be some sort of validation prior to the updating?
Users	EJB JAR	Database communication	See Below		
			AddUser	Entity Bean	Call system database component to add the users to the database
			EditUser	Entity Bean	Call system database component to get the specific user information as per passed requirements
			DeleteUser	Entity Bean	Pass the user information to the system component for deletion; route through master controller for Delete Validate Dialog
ImportData	EJB JAR	Database communication Data Validation	See Below		
			GetData	Session Bean	Get data from external source; must comply with type requirements in XML

			ProcessData	Entity Bean	Process the data according to business rules for specific type defined in XML
			SaveData	Entity Bean	Save data into the database by calling the appropriate
ExportData	EJB JAR	Database communication	See Below		
			GetData	Session Bean	Identify data to be exported from data held within client session; data type requirements are outlined in XML
			SaveData	Entity Bean	Save data in appropriate format as indicated by user selection in controller servlet
Logs	EJB JAR	Database communication	See Below		
			GetLog	Session Bean	Pass the log parameters (from XML) to the database component to retrieve the log information
Backup	EJB JAR		See Below		
			See Exporting Data		
Restore	EJB JAR		See Below		
			See Importing Data		
RestartSystem	EJB JAR		See Below		
			TBD		
SetEnvironment	EJB JAR	Database communication	See Below		
			GetVariables	Session Bean	Get the environment variables from the database and populate into the form
			ModifyVariables	Form Bean	Submit the modified information to the database component for storage in the database.
Routines	EJB JAR	Database communication	See below		
			RunRoutines	Entity Bean	Perform routines based on schedule set

					in SetEnvironment; halts all other application activities until completed.
--	--	--	--	--	--

System Components

Name	Type	Component Relationships and Dependencies	Component	Type	Description
Database Communication (DC1)	EJB JAR	None	See below		The controller servlet for this module; This servlet controller utilizes the Struts 2.0 class library to communicate with EJBs and other application components. Componnets listed underneath this controller are called by controller Execute methods.
			OpenDatabase	Entity Bean	Open a database connection
			CloseDatabase	Entity Bean	Close a database connection
			DatabaseAction	Entity Bean (ActionClass)	Send appropriate parameters of database with data to DataValidation
Transaction Processing (TP1)	EJB JAR	Database Communication	See Below		
			NOTE SURE HOW THIS IS GOING TO WORK. WANT TO USE THE JAVA TRANSACTION API; NEED TO LOOK INTO THIS		
			ScheduledActivity	EntityBean	Look daily for "good until cancel" and "limit orders" to process; create new, unique ID for each of these orders but keep original order ID for processing with FIX gateway
Batch Processing (BP1)	EJB JAR		See below		
			CreateOrders	EntityBean	Create individual orders by looping through bulk data passed; assign unique IDs for each within the array and pass to

					Transaction Processing
			SubAccountProcess	Entity Bean	If a flag comes along with bulk order data that indicates it is a SubAccount order, then after all of the array has been parsed into individual orders, send to the SubAccounting JAR for distribution before processing
Outside Feed Processing (OF1)	EJB JAR	Java Messaging API	See Below		
			GetData	Entity Bean	Get data from outside source; this data could go into temporary datatables; will checking be regular?
			ScrubData	Entity Bean	Validate and scrub data against XML business requirements
Credential Processing (CP1)	EJB JAR	Database Communication	See Below		
			CheckCredentials	Entity Bean	Pass username and password to database and return data with ActionForward to either ApproveCredentials or DenyCredentials.
			ApproveCredentials	Entity Bean	Set user session as approved and return to requesting component
			DenyCredentials	Entity Bean	Set user session as denied and return to requesting component
Database Processing (DP1)	EJB JAR	Database Communication	See Below		
			DeleteData	Entity Bean	Delete data from the data source; data is actually deleted from active tables and saved into a mirrored table with the name "deleted"
			AddData	Entity Bean	Add data to the database; depending upon where the data is coming from (what part of the application) and into what table it's going, additional data can be appended (from XML file)
			UpdateData	Entity Bean	Update data in the database; in the case

					of some data (i.e., orders), updating is actually creating a new record. This logic will be in both the component and the XML file (that provides specification information about what type of data can be updated)
Data Validation (DV1)	EJB JAR	None	See Below		
			ValidateData	Entity Bean	Validate data from DatabaseCommunication against XML business rules for specific data types to ensure proper placement into database; send to DatabaseProcessing component depending upon action requested.
			ValidateDataForm	Entity Bean	Validate form data submitted from application components against XML criteria
Data Calculation (DC1)	EJB JAR	None	See Below		
			Calculate	Entity Bean	Perform calculations on passed data according to XML business rules; return data to requester
Application Errors (AE1)	EJB JAR	Database Communication	See Below		
			MapError	Entity Bean	Map system error against XML errors to determine translated error; call store error
			StoreError	Entity Bean	Store error into the database; call SendError
			SendError	Entity Bean	Send error back to the master servlet for display to the client
Graphing (GR1)	EJB JAR	Third-party graphing API	See Below		
			API Call TBD	Entity Bean	Call to graphing API passing the appropriate parameters and receiving the results

SubAccounting	EJB JAR	Database communication Database processing Batch processing	See Below		
			ScheduledActivity	Entity Bean	Perform scheduled activity according to parameters set in SubAccount. This system component must be instantiated upon application startup; call included in Master Servlet; scheduled activity calls DistributeActivity.
			DistributeActivity	Entity Bean	Distribute apportioned activity across all subaccount portfolios prior to sending to Transaction Processing

JSP Pages

Name	Description
OrderBuy.jsp	Enter a buy order for a client portfolio.
OrderSell.jsp	Enter a sell order for a client portfolio.
OrderEdit.jsp	Edit a sell or buy order for a client portfolio
Login.jsp	Login page
index.jsp	Main page for each module (there is a master index page for the entire application and individual pages for the discreet modules)
Report.jsp	Display reporting information; I am thinking a single page for all of these report types dynamically created.
Export.jsp	Select type of data to export; submit button exports data
Portfolio.jsp	Display a portfolio
PortfolioForm.jsp	Create or edit a portfolio
User.jsp	Depending upon the controller action, display the user page for viewing, editing, or deleting; instantiates the appropriate component from the Users EJB-JAR
Import.jsp	Form for importing data to the system
SystemProperties.jsp	Form for changing the system properties; I am thinking that this will be a dynamic way to adjust the XML files that drive all of the component business logic
Analyze.jsp	Analyze portfolio data
SubAccount.jsp	Create/edit a subaccount

BulkOrderBuy.jsp	Same as OrderBuy.jsp but includes specific information for a bulk purchase (i.e., portfolio drop-down to include, identify sub-account if necessary, etc.)
BulkOrderSell.jsp	Same as OrderSell.jsp but includes specific information for a bulk sale (i.e., portfolio drop-down to include, identify sub-account if necessary, etc.)

