ARMY RESEARCH LABORATORY

Contract Req
Detailed Design Report Version 1:
A C-BASS Component

by Denis McGurin

ARL-MR-459 September 1999

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-MR-459 **September 1999**

Contract Req Detailed Design Report Version 1: A C-BASS Component

Denis McGurin
Corporate Information and Computing Center, ARL

Abstract

This document is the third in a series of reports documenting the structured software engineering of Contract Req Version 1.0. "Contract Req Software Development Plan" and "Contract Req Software Requirement Analysis" are the first two titles in the Contract Req technical documentation library. As part of the Corporate Business Application Software System (C-BASS) suite of work flow applications, Contract Req automates preparation of a requestor's procurement package as part of the contracting process for the U.S. Army Research Laboratory (ARL). Three major sections of this report give background for the Contract Req project, explain the structured methodology, and define the design environment. A fourth segment presents a detailed design for Contract Req using the formalisms of structure charts for defining operations and module specifications for defining objects.

Table of Contents

		Page
	List of Figures	v
	List of Tables	vii
1.	**Introduction**	1
1.1	Contract Req Prototype	1
1.2	Antecedent Documents and Contents of This Report	2
2.	**Structured Design Methodology**	2
3.	**High-Level Requirements and Design Overview**	4
3.1	Lotus Notes Development Environment	4
3.2	Legacy Efforts	6
3.3	Major System Components	6
4.	**Operations Overview**	7
4.1	Human-Machine Interactions	7
4.1.1	*Startup Screen*	8
4.1.2	*Contract Request Data Entry Screen*	8
4.1.3	*Approval Screen*	10
4.1.4	*Status Inquiry Screen*	12
4.1.5	*Reports Screen*	12
4.2	Software-to-Software Interactions	12
4.2.1	*SOMARDS*	13
4.2.2	*Lotus Notes Control Agents*	14
5.	**Procedural Design Description**	15
5.1	System Navigation	15
5.2	Prepare Contract Request	20
5.3	Process Pending Contract Request	23
5.4	Obtain Approvals	25
5.5	Edit Contract Request	29
5.6	Status Inquiry	32
5.7	Generate Report	33
5.8	Legacy System Interfaces	34
5.9	Certify Funds	39
5.10	Lotus Notes Agent Processes	42

Page

6. **References** 45

Distribution List 47

Report Documentation Page 49

List of Figures

Figure		Page
1.	Overview of Contract Req System Design	7
2.	Contract Req Startup/Inbox User Screen.	9
3.	Data Entry Screen, Top Portion	10
4.	Data Entry Screen, Middle Portion	11
5.	Data Entry Screen, Bottom Portion	12
6.	Status Inquiry Screen	13
7.	Reports Screen	14
8.	Contract Req Major Components and User Interface Navigation	17
9.	Prepare Contract Request	21
10.	Process Pending Contract Request	24
11.	Obtain Approvals	26
12.	Edit Contract Request	30
13.	Status Inquiry	32
14.	Generate Report	33
15.	Legacy System Interfaces	34
16.	Certify Funds	39
17.	Lotus Notes Agent Processes	42

INTENTIONALLY LEFT BLANK.

List of Tables

Table		Page
1.	Overview of Structure Charts and Module Specifications.	16

INTENTIONALLY LEFT BLANK.

1. Introduction

As the demand for cost-effective R&D capabilities increases, the U.S. Army Research Laboratory (ARL) looks to modern information technology (IT) as an enabler for core organizational practices. As part of a general trend toward streamlining and automating critical transactions, the Corporate Business Application Software System (C-BASS) is a suite of software applications characterized by

- Robustness and scalability,
- Maximization of commercial off-the-shelf (COTS) products,
- Compatibility with standard systems,
- Focus on interoperability and elimination of "stovepipes," and
- Emphasis on security.

1.1 Contract Req Prototype. As a component of C-BASS, the purpose of the Contract Req prototype is to model a secure client/server system for processing specific portions of the contracting process. (Throughout this document, "Contract Req" is used as an abbreviated form of Contract Request Version 1.0.) The motivating force behind this work has been ARL down-sizing and the findings put forth in the Business Process Reengineering (BPR) report on the contracting process. The BPR "To-Be Model: Formal Contracts" [1] identifies potential process improvements (some of which require computer automation) that would increase productivity of contracting operations at ARL.

The proof-of-principle Contract Req will alleviate some of the risks involved in implementing new technologies used to build the emerging ARL Intranet. As currently envisioned, ARL Intranet capabilities will consist of an integrated set of Lotus Notes and Web-based software to support research and development (R&D) competencies. Such a large and complex undertaking requires both careful planning and proactive management. Additionally, the Contract Req software development will help to refine requirements described in the ARL BPR "To-Be Model" [1]. Development of a full production Contract Req will proceed in phases,

using an incremental, evolutionary approach that emphasizes integration of "lessons learned" from the prototype design, implementation, and evaluation.

1.2 Antecedent Documents and Contents of This Report. Two documents precede this report: "Contract Req Software Development Plan" [2] and "Contract Req Software Requirements Analysis" [3]. The first gives an overview of the system's purpose, identifies organizational requirements and constraints, develops a management plan, and sets forth a timetable with major milestones. The second document elaborates on Contract Req design using the formalisms of data flow diagrams, structured English narratives of the system's functionality, and a data dictionary.

This document contains four sections:

- "Structured Design Methodology" - briefly explains the methodology used to extract the software functional specifications.

- "High-Level Requirements and Design Overview" - describes the design opportunities and constraints, identifies a software reuse strategy, and presents the basic features of the Lotus Notes development environment.

- "Operations Overview" - presents operational scenarios, including user interfaces and interaction with legacy systems, as well as core components of Lotus Notes.

- "Procedural Design Description" - breaks the system design into finer detail and presents structure charts and module specifications.

2. Structured Design Methodology

Modern software engineering relies on structured analysis to design coherent, reliable, and maintainable data processing systems [4]. Using a process of step-wise refinement, the designer

starts with a top-level, conceptual architecture for the system and moves downward, incrementally decomposing each part and providing greater specificity for each functional unit. The purpose of a structured design approach is to generate a detailed description of how to build software objects that functionally adhere to the structured system analysis models and meet all technical system requirements.

As a major element in an application's documentation library, the detailed design document gives functional specifications at the granularity required for a programming language implementation of each component. In other words, at this level of decomposition, the detailed design (also called the procedural design) defines each operation and each object within the system.

Within the software engineering process, a detailed design document serves three purposes. First, the design is a product review, used to uncover logic errors prior to programming investment. Second, a codified detailed design serves as a template to guide coding. Third, the document helps with the derivation of test cases for software validation.

As the procedural design document for Contract Req prototype, this report finalizes the design of all data structures internal to each component and provides a representation of program logic at a relatively low level of abstraction. Two important representations used in this document are structure charts and module specifications. The structure chart shows the division of a system into modules, while also displaying their hierarchical relationship, organization, connections, and interaction with one another. The module specifications give detailed information on how units process the interface inputs in order to produce the necessary outputs. A Contract Req representation using these formalisms is given in section 5, "Procedural Design Description."

The following section reviews the design opportunities and constraints that resulted in the conceptual model of Contract Req. Section 4, "Operations Overview," takes a transactional approach to describing the system. First, Contract Req is viewed from the perspective of the end

user by summarizing screen design and human-machine interaction. Then Contract Req's transactions a legacy system—Standard Operations and Maintenance Army Research and Development System (SOMARDS)—are delineated. The last portion of section 4 covers Contract Req's transactions with the core capabilities of Lotus Notes.

3. High-Level Requirements and Design Overview

Requirements for the C-BASS suite and its components were accrued from three sources: (1) general organizational streamlining and business process re-engineering, (2) user needs conjoined with "best" practices, and (3) the opportunities and the limitations provided by existing hardware and software. Of special interest are both the features offered by Lotus Notes for groupware development and the issues stemming from legacy systems.

3.1 Lotus Notes Development Environment. Drawing momentum from the climate of organizational change, the Corporate Information and Computing Center (CICC) has designated computer-supported collaborative work (CSCW) applications as a cornerstone in ARL's emerging Intranet. As determined by a recent ARL task force, critical features for a viable Intranet computer architecture include

- Uniformity of communication among and within ARL sites,
- Use of COTS products,
- Adherence to standards (American National Standards Institute [ANSI], Department of Defense [DOD], and de facto) wherever possible,
- Multiplatform support,
- Adequate security,

- Support for workflow operations to improve business processing, and

- Modern data management capabilities.

Lotus Notes was selected as a CSCW enabling technology because the product provides three invaluable capabilities:

- Potential for systematic reuse of applications built in LotusScript,

- A commercially viable, large-scale product with enough market share to ensure continued evolution, and

- A complete development environment that provides generic/adaptable core modules for standard business practices.

Lotus Notes core components include a uniform user interface, primary data store capabilities, and workflow routing using a robust messaging engine. Additionally, the developer's library of generic applications includes a simple approval/routing example. Reuse of this system's basic logic reduces design and implementation effort, requiring only minimal expansion effort in the number of approval cycles to handle the approval and routing of contract requests.

Through the Lotus Notes "control agent" feature (similar to a "macro" in other software), developers can customize routine functionality to create powerful tools that reflect preferences in handling work. The "control agent" captures an event (or set of events) within the operation and triggers a subsequent action. For example, an agent may sort incoming email and place it into appropriate folders after being trigged by specified keywords in the subject field. Other agents may tailor database queries or filter and route results. Still others may look for specified events and then trigger a response, such as an automated reminder sent to an email list server.

3.2 Legacy Efforts. During the BPR development effort of 1995, the former Enterprise Systems Branch (ESB) fielded an on-line interface to SOMARDS. The interface pulled purchase request funding data from an Oracle database and then used Expect (a software package automating interactive processes) to connect to the SOMARDS user interface and certify funds on-line. Reuse of this interface package requires only minor changes for the Contract Req prototype.

3.3 Major System Components. As illustrated in Figure 1, at a high level of representation, Contract Req contains four major components:

- User Interface - consisting of a set of input/output (I/O) screens, this process allows requesters to enter data, functional users to approve requests, and all users to perform status inquiries and generate reports. The user interface is implemented using the Lotus Notes client, located on the user's desktop workstation.

- Data Store "CONTRACT REQUESTS" - implemented on the central Lotus Notes server, this entity contains all the active, canceled, and closed contract requests that are or have been processed by Contract Req.

- Legacy System Interfaces - this process connects to the SOMARDS system for fund certification and for instantiating contract orders. Portions of the legacy system interfaces will be implemented on a Unix server and an Oracle database, with connects to the Notes server through the network.

- Control Agents - analogous to an event-oriented application, this process receives signals (called "triggers") from the user interface, which in turn controls the timing of the legacy system interactions.

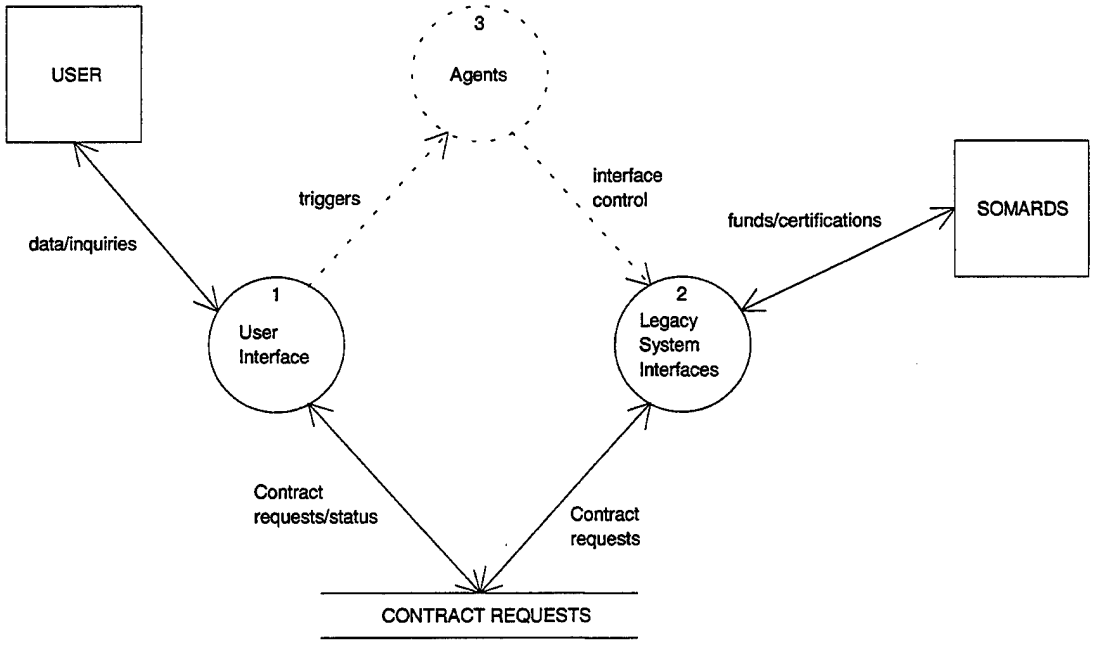


Figure 1. Overview of Contract Req System Design.

4. Operations Overview

In addition to showing the overall design of Contract Req, Figure 1 also helps identify the system interaction scenarios. Representing these scenarios are graphics of the user interface (the screens options described in following text) and the background processes that communicate with the legacy system.

4.1 Human-Machine Interactions. Modern design of user interfaces includes highly visual menu options for selecting processes, iconic representations for system features, and forms displays for data I/O and information consolidation. All major functionality for Contract Req users exists within five central interface screens: (1) Startup, (2) Data Entry, (3) Approval, (4) Status Inquiry, and (5) Cycle-Time Report. Figures 2 through 7 more fully illustrate human-machine interactions indicated on each screen.

4.1.1 Startup Screen. When a user first logs into Contract Req, the startup (or inbox) screen shown in Figure 2 appears. The large area on the right is the user's inbox, where requests requiring attention are listed. The inbox contents are based on the individual user's role. When the user selects a particular request, the system displays the entire contract request and allows the user to perform processing (e.g., approval, data entry). Along the left-hand side of the inbox screen are buttons by which the user navigates to the various functional areas of Contract Req. Four functions are indicated:

- Create New Request - initiates a new contract request; fills in the requester information; displays the contract request entry screen,

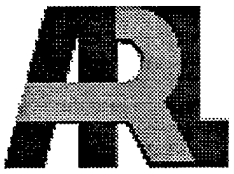
- Pending - displays or refreshes the inbox screen,

- Status - determines the current status of contract requests that have been "touched" by the user and displays results on the status inquiry screen, and

- Reports - generates the cycle time report and displays on the reports screen.

4.1.2 Contract Request Data Entry Screen. The "Contract Request Entry" screen is shown in Figures 3–5. Lotus Notes does not limit the length of a screen; thus, screens that are both longer than the size of a screen and longer than the size of a piece of paper can be developed. Therefore, the entry screen can be mostly self-contained and take more than a single page to depict. Based on the userid of the requester, the request will be filled in with the requester's name, office symbol, phone, and address. The requester then fills in the rest of the request information (e.g., date required, priority, vendors, items, fund source, etc.). By pressing the appropriate button along the top of the screen, the requester can

- cancel the request,
- save the request for later data entry, or
- submit the request to their supervisor for approval.



Contract Req

Create New Request	Doc Ref No.	Requestor	Request Date
	C-W71B7J-80016-A001	Fred Smith	16 Oct
Pending			
Status			
Reports			

Figure 2. Contract Req Startup/Inbox User Screen.

Contract Request

Requestor Info:
Name:
Telephone Number:
Brief Description:

Organization:
Date Requested:

Delivery Date or Period of Performance:

Statement of Work

SOW Status

SOW Help

Suggested Vendor Information:
Vendor Name:
Address:
Fax Number:

Point of Contact:
Phone Number:
Business Type:

Market Research:

MR Status?

Market Research Help

Figure 3. Data Entry Screen, Top Portion.

4.1.3 Approval Screen. The "Approval" screen is the same screen as the data entry screen. The difference in the two is the addition of buttons at the top of the screen to approve the request, deny the request, or close and defer judgment. This screen is displayed when the user selects a request from the inbox. Full contract request details are displayed for the approving official and, depending on his or her role, the official may have editing privileges for some of the fields:

- supervisor - full editing rights on all fields,
- budget analyst - editing of fund source,
- property book officer - item tags, and
- contracting officer - buyer assignment.

Once the user has reviewed the request, he can press the appropriate button to

- close the display and save approval for later,
- approve the request, or
- deny approval and enter an explanation.

Government Cost Estimate: (Not required for BAA, SBIR)

GCE Help

Item	Direct Labor	Hours		Unit Cost	Total Cost
	DLTotal				
	Labor Overhead			% of Total DL	
	Other Direct Costs:				
	Direct Materials				
	Subcontract(s)				
		DM & Subkt Subtotal			
	Material Handling			% of DM & Subkt	
	Travel Expense				
	Consultant		Hrs	atPer	
	Misc ODC's				
				SubTotal	
	General & Administrative Expense				%
				SubTotal	
	Profit / Fee			%	
	Total Estimated Price/Cost				

Figure 4. Data Entry Screen, Middle Portion.

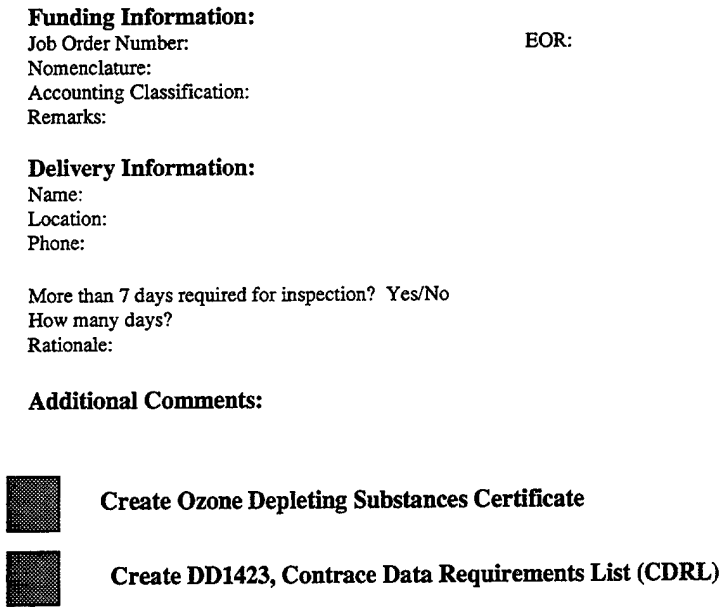


Figure 5. Data Entry Screen, Bottom Portion.

4.1.4 Status Inquiry Screen. The "Status Inquiry" screen is shown in Figure 6. This screen displays the requests that have been "touched" by the user, either as requester or approving official, and what the current status is. By selecting a request, the full details will be displayed as shown in the "Approval" screen (see Figures 3–5), but without the "Approve" and "Deny" buttons. Additionally, editing the fields will not be allowed in the status inquiry mode.

4.1.5 Reports Screen. Figure 7 shows the "Reports" screen. This screen displays the "Cycle Time Report," which computes the average time a request waits for approval in the various functional areas. Navigation choices along the left side of the screen will allow the user to select different reports as they become available.

4.2 Software-to-Software Interactions. In addition to the user-initiated interactions, Contract Req's background processes communicate both with the legacy system (SOMARDS) and with the groupware functions provided in the Lotus Notes development environment. Each is more fully described in the following paragraphs.

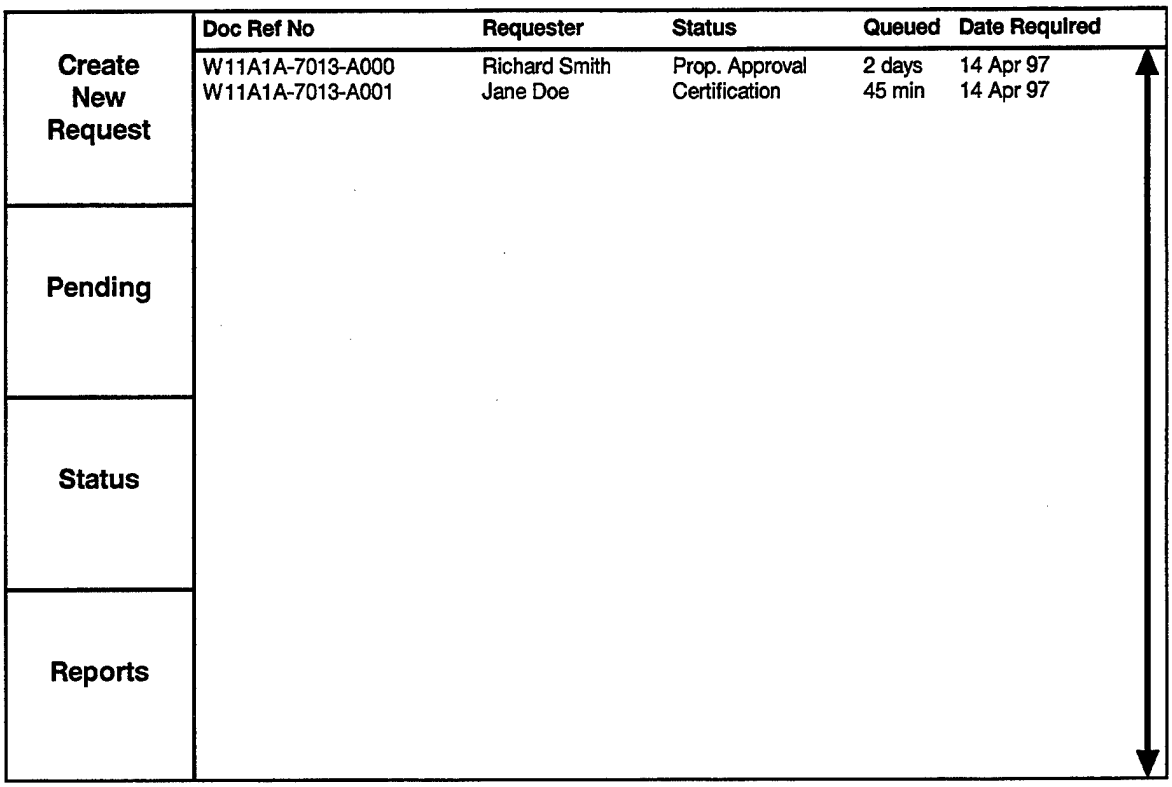


Figure 6. Status Inquiry Screen.

4.2.1 SOMARDS. The SOMARDS interface operates in three modes: (1) build a transaction block, (2) certify funds, and (3) reconcile the transaction block. Each operation uses the SOMARDS on-line user interface. Building a block occurs first thing in the morning, after SOMARDS is brought up. The block is named and the initial balance set to zero. This transaction block is then used during the course of the day for all certifications processed by Contract Req. As submitted requests are deemed eligible for certification, they are placed in a first-in-first-out (FIFO) queue. The certification process continually checks the queue and processes the "certifiable requests" as they become available. Depending on the number of pending transactions, the response time could be a few seconds or a few minutes.

Certification results are placed in a second queue, which is checked by a Lotus Notes control agent. This processing element routes certified requests to their next destination, while rejected requests are sent back to the requester for indicated corrections. Reconciliation occurs just before SOMARDS is brought down at the end of the day. The cumulative total is read from

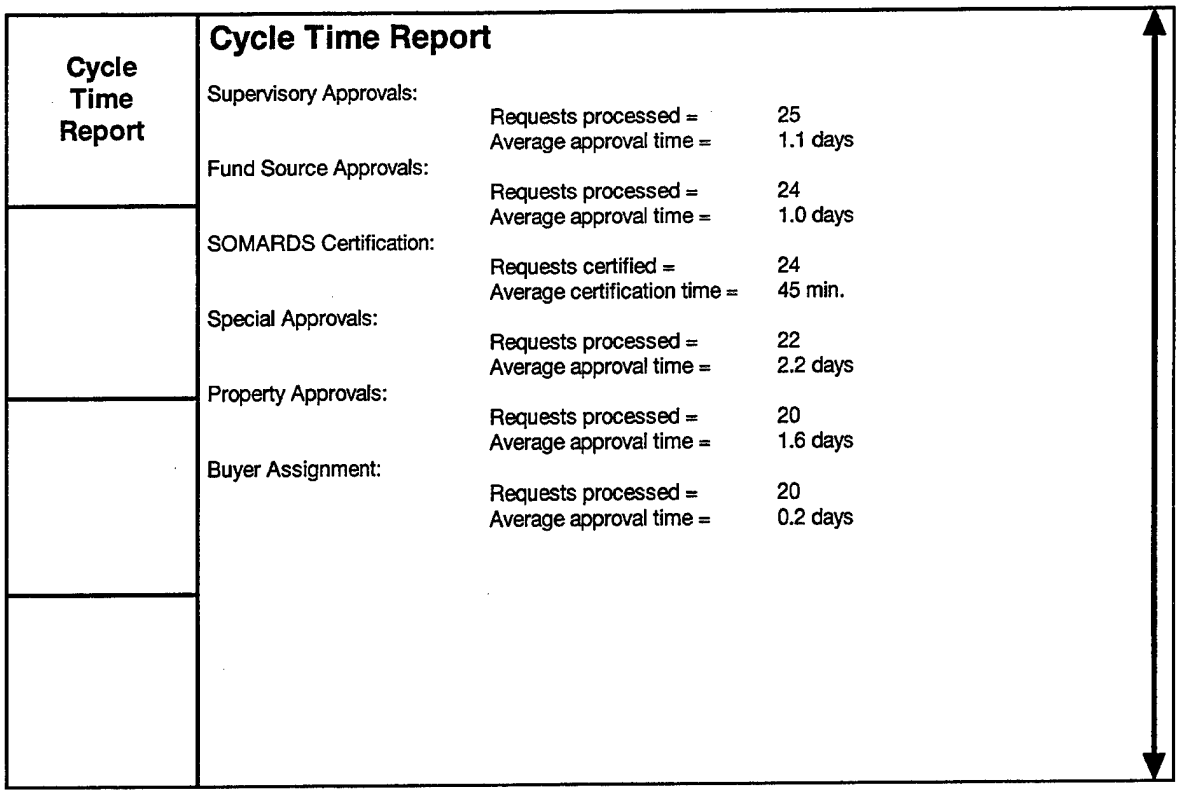


Figure 7. Reports Screen.

SOMARDS and checked against a running total calculated from Contract Req. Any discrepancies are sent to the Contract Req administrator, and the block is reconciled using the SOMARDS cumulative value.

4.2.2 Lotus Notes Control Agents. The control agents reside on the Notes server and operate in two modes: either triggered by the user interface client or automatically executed at prescribed intervals.

Triggers from the user interface consist of (1) fund source or actual cost approvals (SOMARDS certification) or (2) a contract specialist assignment. The funding approval event activates a Notes control agent, which puts the information required by SOMARDS into a FIFO queue. The contract specialist assignment action triggers a different Notes control agent that puts the request information into a separate FIFO queue. Both queues reside in an Oracle database and are processed as described previously.

Using server-based control agents achieves a significant economy. If the user interface alone were used to communicate with the queues, the Oracle database network connection software (SQL*Net) would be required for each client. Thus, locating the agents on the server and triggering them from the user interface saves the cost and administration of having the extra client software.

In addition to being event-driven, these Lotus Notes server agents also execute at prescribed intervals. Whether event triggered or interval initiated, the control agents process information received from the legacy systems, as illustrated in this example. The SOMARDS fund certification agent interprets the message returned from SOMARDS and marks the request either as certified or rejected. If certified, the request is routed to the next destination. If rejected, an error message is placed in the "explanation" field and the request is returned to its originator.

5. Procedural Design Description

At the procedure level of a structured design document, the information system is partitioned into modules and transactions among these elements are defined. Within this section, two formalisms are used to give a detailed design specification for Contract Req. First, structure charts provide a graphic representation of the modules and their hierarchical structure. These charts also show the data/information transactions and central interfaces among the modules. Second, specifications for each procedural component (module) are given using four categories of definition: purpose, uses, returns, and functional details.

Table 1 is a composite of Contract Req, indicating 10 functional components, a decomposition of each into its associated modules, and the figure number for the structure chart depicting each functional component as a cluster of modules and interactions.

5.1 System Navigation. The major components of the Contract Req are shown in the structure chart displayed in Figure 8. The "Navigation" module represents the user interface, and the "LI" continuation marker represents the legacy system interfaces (which are shown in more

Table 1. Overview of Structure Charts and Module Specifications

Functional Component	Associated Module Specifications	Structure Chart
Navigation	- Contract Req - Navigation - Show inbox - Get Pending Contract Requests - Select Action	Figure 8
Prepare Contract Request	- Prepare Contract Request - Create New Contract Request - Fill in Contract Request - Fill in Fund Source	Figure 9
Process Pending Contract Request	- Process Pending Contract Request - Cancel Contract Request	Figure 10
Obtain Approvals	- Obtain Approvals - Approve Contract Request - Approve Fund Source - Approve Actual Cost - Queue Funding - Approve Special Items - Approve Property	Figure 11
Edit Contract Request	- Edit Contract Request - Edit Rejected Contract Request - Get Item Tags - Assign Contract Specialist	Figure 12
Status Inquiry	- Status Inquiry	Figure 13
Generate Report	- Generate Report	Figure 14
SOMARDS Legacy System Interfaces	- Legacy Systems Interfaces - Connect to SOMARDS - Build Block - Send Block Data - Get Build Return Message - Reconcile Block - Send Reconcile Info - Get Reconcile Return Message - Disconnect from SOMARDS	Figure 15
Certify Funds	- Certify Funds - Get Next Entry from Queue - Send Funds Info - Get Certify Return Message - Update Daily Balance	Figure 16
Lotus Notes Agent Processes	- Agents - Certification Queue Agent - SOMARDS Return Agent	Figure 17

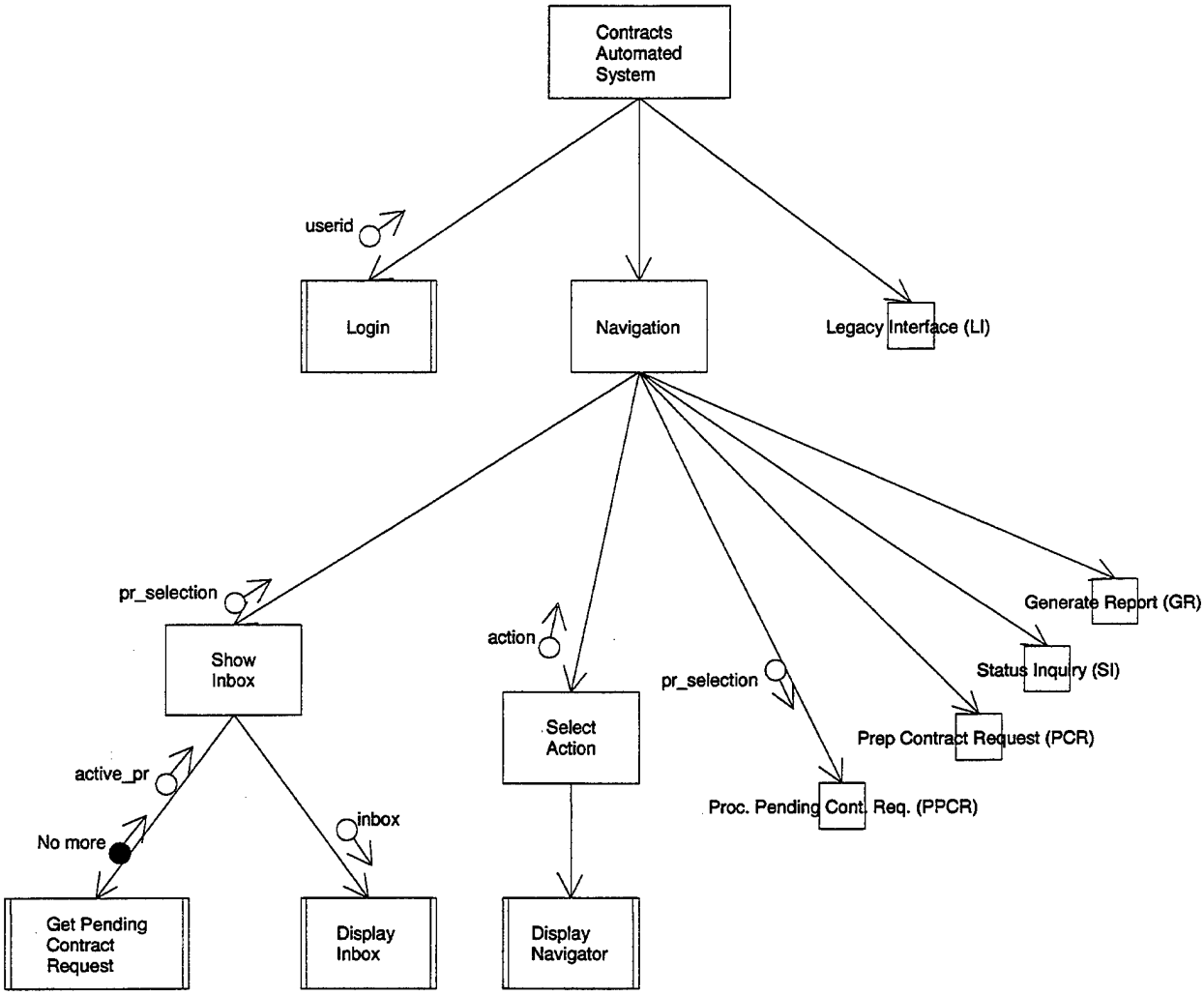


Figure 8. Contract Req Major Components and User Interface Navigation.

detail later). The continuation markers (e.g., "PCR", "PPCR", "SI", and "GR") point to the structure charts (shown later in the text) that represent the other functional areas of the user interface.

The module specifications associated with Figure 8 are listed as follows:

Module: Contract Req.

Purpose: Provide a user interface for contract request processing and an automated interface to legacy systems.

Uses: Login, Navigation, Legacy Interfaces.

Returns: N/A.

Functional Details:

1. Initiate Agents.
2. Initiate Legacy Interfaces.
3. Wait for open database attempt and get Login.
4. If login a success, start Navigation user interface.

Module: Navigation.

Purpose: Allow user access to the various functional areas of the user interface.

Uses: Show Inbox, Select Action, Prepare Contract Request, Process Pending Contract Request, Status Inquiry, Generate Report.

Returns: N/A.

Functional Details:

1. Wait for Show Inbox or Select Action to return pr_selection or action.
2. If pr_selection is not null, then Process Pending Contract Request using pr_selection.
3. If action is not null, then, depending on action selected, Prepare Contract Request or Status Inquiry or Generate Report.

Module: Show Inbox.

Purpose: Display to user contract requests requiring his or her attention and allow user select a particular request for processing.

Uses: Get Pending Contract requests, Display Inbox.

Returns: pr_selection.

Functional Details:

1. Get Pending Contract requests until no more pending requests.
2. Display Inbox using pending requests.
3. Wait for user to select a particular request.
4. Return pr_selection to parent module.

Module: Get Pending Contract requests.

Purpose: Retrieve contract requests requiring the user's attention.

Uses: N/A.

Returns: active_pr.

Functional Details:

1. Scan the ACTIVE data store for requests with user or role set as the author.
2. Return the active_pr or control signal if no more.

Module: Select Action.

Purpose: Allow user to select a particular functional area of the user interface.

Uses: Display Navigator.

Returns: action.

Functional Details:

1. Display Navigator.
2. Wait for user to select action using navigator buttons.
3. Return selected action to parent module.

5.2 Prepare Contract Request. Figure 9 is the structure chart for the "Prepare Contract Request" process. This module creates a new contract request and allows the requester to fill in necessary data elements.

The module specifications associated with Figure 9 are listed as follows.

Module: Prepare Contract Request

Purpose: Allow requester to fill in a contract request and submit it for approval.

Uses: Create new Contract Request, display Blank Contract Request, Fill in Contract Request, Fill in Fund Source.

Returns: funded_pr.

Functional Details:

1. Create New Contract Request using the userid.
2. Display Blank Contract Request for user input.
3. Fill in Contract Request and Fill in Fund Source.
4. Wait for user to submit, save, or cancel new contract request.
5. If canceled go to inbox screen.
6. If save, store contract request in ACTIVE.
7. If submit, determine supervisor userid and set request author to supervisor.

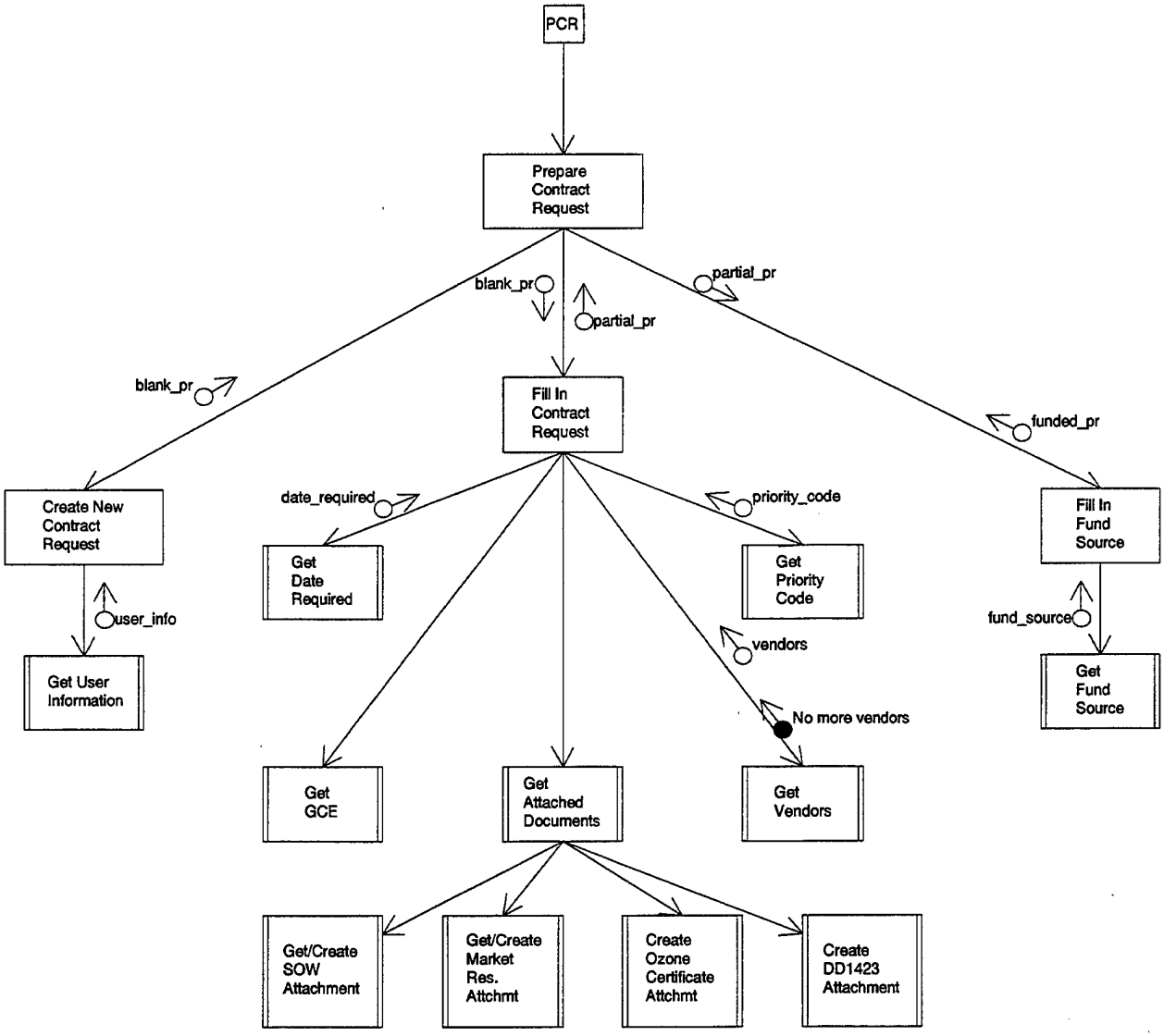


Figure 9. Prepare Contract Request.

Module: Prepare Contract Request

Purpose: Allow requester to fill in a contract request and submit it for approval.

Uses: Create new Contract Request, display Blank Contract Request, Fill in Contract Request, Fill in Fund Source.

Returns: funded_pr.

Functional Details:

1. Create New Contract Request using the userid.
2. Display Blank Contract Request for user input.
3. Fill in Contract Request and Fill in Fund Source.
4. Wait for user to submit, save, or cancel new contract request.
5. If canceled go to inbox screen.
6. If save, store contract request in ACTIVE.
7. If submit, determine supervisor userid and set request author to supervisor.

Module: Create New Contract Request.

Purpose: Create a new contract request with the requester's user_info filled in.

Uses: Get User Information.

Returns: blank_pr.

Functional Details:

1. Get User Information using requesters userid.
2. Set the requester user_info in the new contract request.
3. Return the blank_pr to parent module.

Module: Fill in Contract Request.

Purpose: Fill in the blank contract request.

Uses: Get Date Required, Get Items, Get Vendors, Get Priority Code.

Returns: partial_pr.

Functional Details:

1. Get GCE Items and set in the contract request until no more items.
2. Get Vendors and set in the contract request until no more vendors.
3. Return partial_pr to parent module.
4. Get/Create Statement of Work (SOW).
5. Get/Create Market Research (MR).
6. Create Ozone Depleting Substances Certificate if required
7. Create DD 1423, Contract Requirements Data List (CDRL) if required.

Module: Fill in Fund Source.

Purpose: Fill in the fund source information.

Uses: Get Fund Source.

Returns: funded_pr.

Functional Details:

1. Get Fund Source and set in the contract request.
2. Return funded_pr to parent module.

5.3 Process Pending Contract Request. Figure 10 shows the "Process Pending Contract Request" process structure chart. This module allows users of Contract Req to process contract requests (i.e., approvals, edit, cancel, receive, and accept shipment).

The module specifications associated with Figure 10 are listed as follows:

Module: Process Pending Contract Request.

Purpose: Provide a means for users to process pending contract requests.

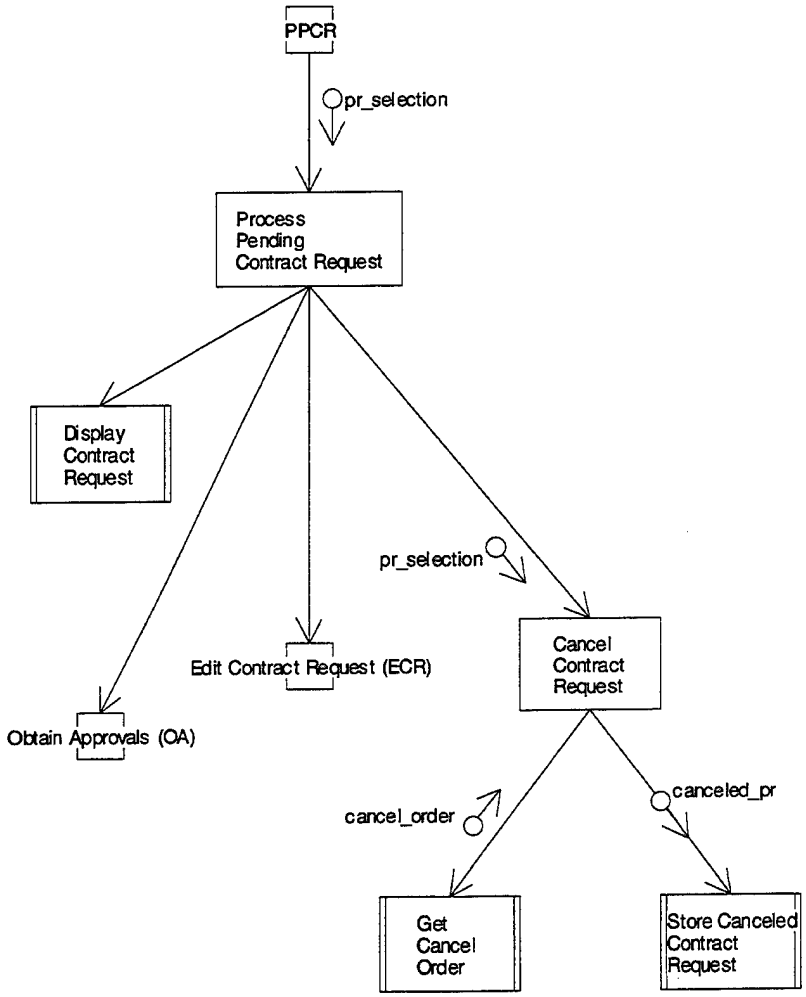


Figure 10. Process Pending Contract Request.

Uses: Display Contract Request, Obtain Approvals, Edit Contract Request, Cancel Contract Request, Receive Order, Accept Shipment.

Returns: N/A.

Functional Details:

1. Display Contract Request to user.
2. If user role matches the contract, request next approval, then Obtain Approvals.

3. If user is the requester and pr_selection is a rejected_pr, then Edit Contract Request.
4. If user role is property and pr_selection is a special_pr, then Edit Contract Request.
5. If user role is contracting officer and pr_selection is a orderable_pr, then Edit Contract Request.
6. If user is the requester or requester's supervisor, then allow Cancel Contract Request.

Module: Cancel Contract Request.

Purpose: Allow the requester or supervisor to cancel the request.

Uses: Get Cancel Order, Store Canceled Contract Request.

Returns: N/A.

Functional Details:

1. Get Cancel Order from user.
2. If cancel order is "Yes," then Store Canceled Contract Request in the CANCELED data sore.

5.4 Obtain Approvals. Figure 11 shows the "Obtain Approvals" process structure chart.

Module: Obtain Approvals.

Purpose: Obtain approvals from the necessary approving officials.

Uses: Approve Contract Request, Approve Fund Source, Approve Actual Cost, Approve Special Items, Approve Property, Approve Contract.

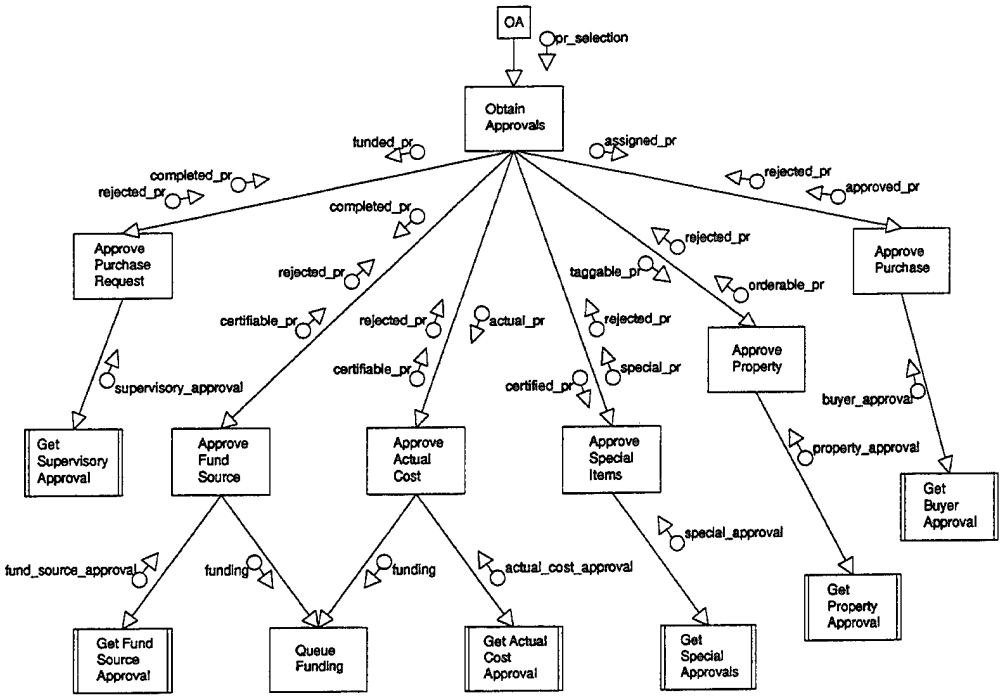


Figure 11. Obtain Approvals.

Returns: N/A.

Functional Details:

1. If user role is supervisor and pr_selection is funded_pr, then Approve Contract Request.
2. If user role is budget analyst and pr_selection is a completed_pr, then Approve Fund Source.
3. If user role is supervisor and pr_selection is a actual_pr, then Approve Acutal Cost.
4. If user role is special approver and pr_selection is a certified_pr, then Approve Special Items.
5. If user role is property book and pr_selection is a taggable_pr, then Approve Property.

6. If user role is contract specialist and pr_selection is an assigned_pr, then Approve Contract.
7. Store approved or rejected contract request in the ACTIVE data store.

Module: Approve Contract Request.

Purpose: Obtain supervisory approval.

Uses: Get supervisory Approval.

Returns: completed_pr, rejected_pr.

Functional Details:

1. Get Supervisory Approval from supervisor.
2. If approval is "Yes," then determine budget analyst and set to author.
3. If approval is "No," then set author to requestor.

Module: Approve Fund Source.

Purpose: Obtain fund source approval from budget analyst.

Uses: Get Fund Source Approval, Queue Funding.

Returns: certifiable_pr, rejected_pr.

Functional Details:

1. Get Fund Source Approval from budget analyst.
2. If approval is "Yes," then Queue Funding.
3. If approval is "No," then set author to requestor.

Module: Approve Actual Cost.

Purpose: Obtain actual cost approval from supervisor.

Uses: Get Actual Cost Approval, Queue Funding.

Returns: certifiable_pr, rejected_pr.

Functional Details:

1. Get Actual Cost Approval from supervisor.
2. If approval is "Yes," then Queue Funding.
3. If approval is "No," then set author to requestor.

Module: Queue Funding.

Purpose: Queue the funding information for certification by the SOMARDS interface.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Trigger certification agent along with doc_ref_no.

Module: Approve Special Items.

Purpose: Obtain approval for special items.

Uses: Get Special Approvals.

Returns: special_pr, rejected_pr.

Functional Details:

1. Get Special Approval form special approving official.
2. If approval is "Yes," then determine property book officer and set to author.
3. If approval is "No," then set author to requestor.

Module: Approve Property.

Purpose: Obtain property approval from the property book officer.

Uses: Get Property Approval.

Returns: orderable_pr, rejected_pr.

Functional Details:

1. Get Property Approval form supervisor.
2. If approval is "Yes," then determine contracting officer and set to author.
3. If approval is "No," then set author to requestor.

5.5 Edit Contract Request. Figure 12 shows the "Edit Contract Request" process structure chart. This module allows users to edit various fields in the contract request based on their role and the current state of the request.

The module specifications associated with Figure 12 are listed as follows:

Module: Edit Contract Request.

Purpose: Allow users to edit a contract request based on their roles and the state of the contract request.

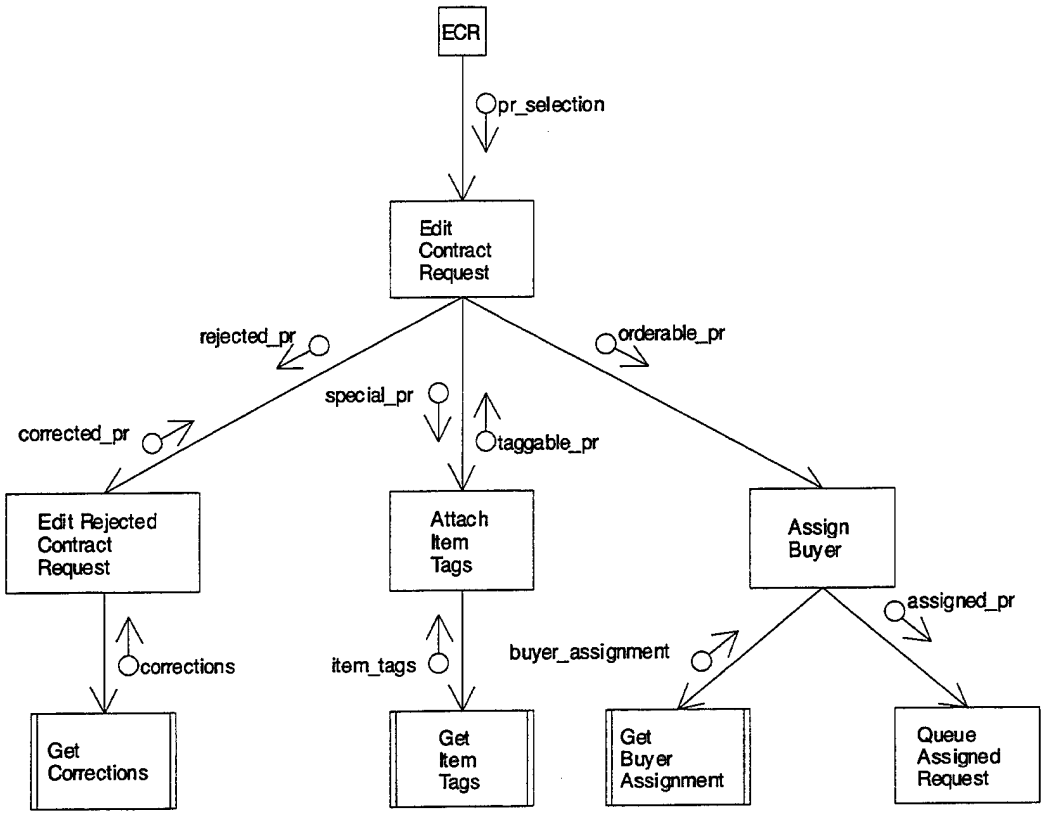


Figure 12. Edit Contract Request.

Uses: Edit Rejected Contract Request, Attach Item Tags, Assign Buyer.

Returns: N/A.

Functional Details:

1. If user is the requester and pr_selection is a rejected_pr, then Edit Rejected Contract Request.
2. If user role is property and pr_selection is a special_pr, then Attach Item Tags.
3. If user role is contracting officer and pr_selection is an orderable_pr, then Assign Buyer.

Module: Edit Rejected Contract Request.

Purpose: Allow the requester to correct a contract request that has been rejected.

Uses: Get Corrections.

Returns: corrected_pr.

Functional Details:

1. Get Corrections from the requester.
2. Clear all approvals and submit for supervisory approval.
3. Return corrected_pr to parent module.

Module: Attach Item Tags.

Purpose: Allow the property book officer to attach item tags.

Uses: Get Item Tags.

Returns: taggable_pr.

Functional Details:

1. Get Item Tags from the property book officer until no more item tags.
2. Return taggable_pr to parent module.

Module: Assign Contract Specialist.

Purpose: Assign a specialist to a particular contract request.

Uses: Get Specialist Assignment, Queue Assigned Request.

Returns: N/A.

Functional Details:

1. Get Specialist Assignment from contracting officer.

5.6 Status Inquiry. Figure 13 shows the "Status Inquiry" process structure chart. This module allows users to check the current status of contract requests that have been "touched" by them.

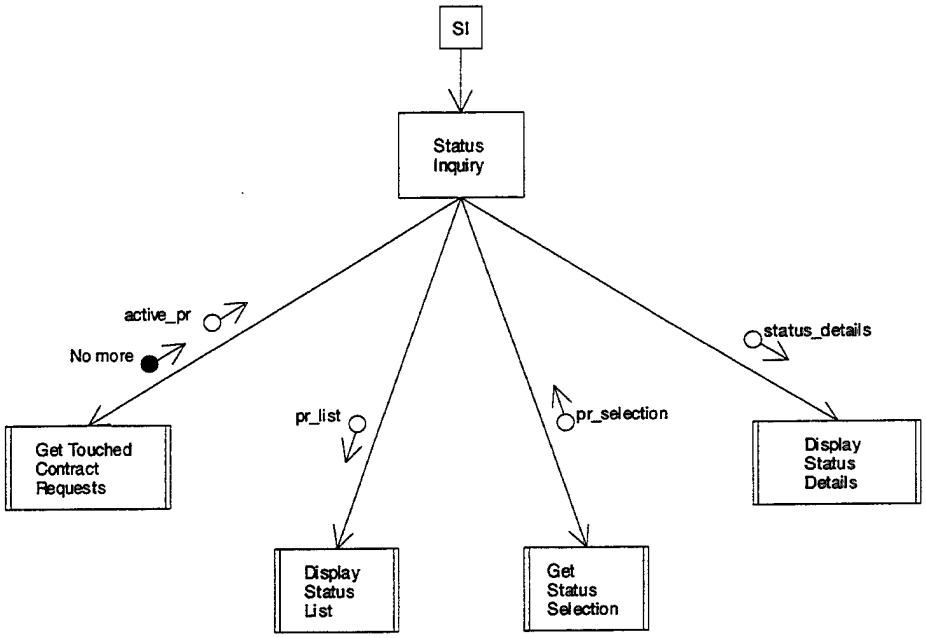


Figure 13. Status Inquiry.

The module specifications associated with Figure 13 are listed as follows:

Module: Status Inquiry.

Purpose: Display the status of requests that have been "touched" by the user.

Uses: Get Touched Contract Request, Display Status List, Get Status Selection, Display Status Details.

Returns: N/A.

Functional Details:

1. Get Touched Contract requests from ACTIVE data store until no more.
2. Display Status List of contract requests to user.
3. Get Status Selection from user.
4. Display Status Details for pr_selection.

5.7 Generate Report. Figure 14 shows the "Generate Report" process structure chart. This module allows users of Contract Req to generate summary reports.

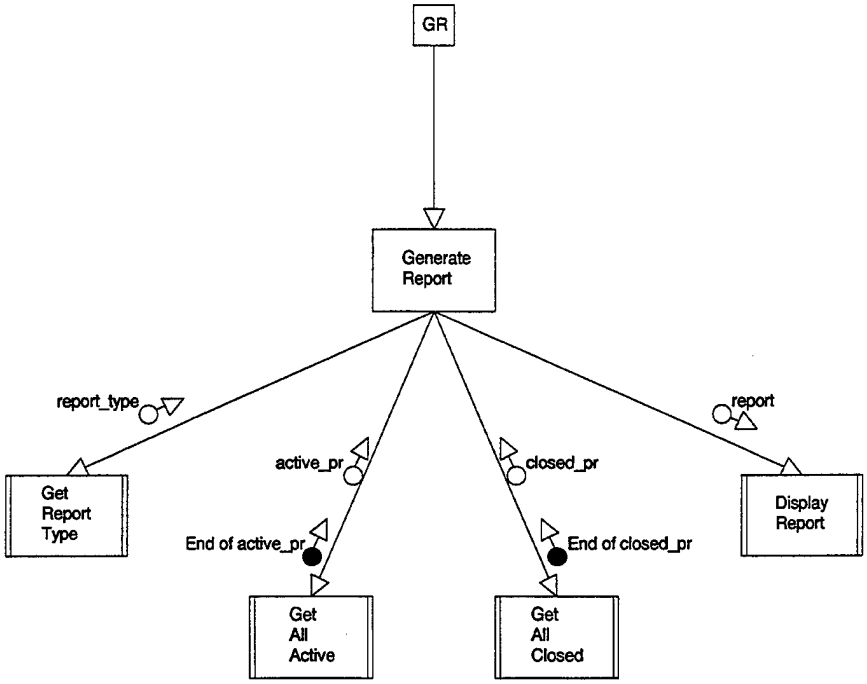


Figure 14. Generate Report.

The module specifications associated with Figure 14 are listed as follows:

Module: Generate Report.

Purpose: Generate various reports that summarize the performance of the small contract process.

Uses: Get Report Type, Get All Active, Get All Closed, Display Report.

Returns: N/A.

Functional Details:

1. Get Report Type from user.
2. Get All Active contract request until no more.
3. Get All Closed contract requests until no more.
4. Calculate summary data depending on report type.
5. Display report to user.

5.8 Legacy System Interfaces. Figure 15 shows the "Legacy Interfaces" process structure chart. This module provides an automated interface between Contract Req and the legacy system SOMARDS.

The module specifications associated with Figure 15 are listed as follows:

Module: Legacy Interfaces.

Purpose: Provide an automated interface between Contract Req and the legacy system SOMARDS.

Uses: Connect to SOMARDS, Build Block, Certify Funds, Reconcile Block, Disconnect from SOMARDS.

Returns: N/A.

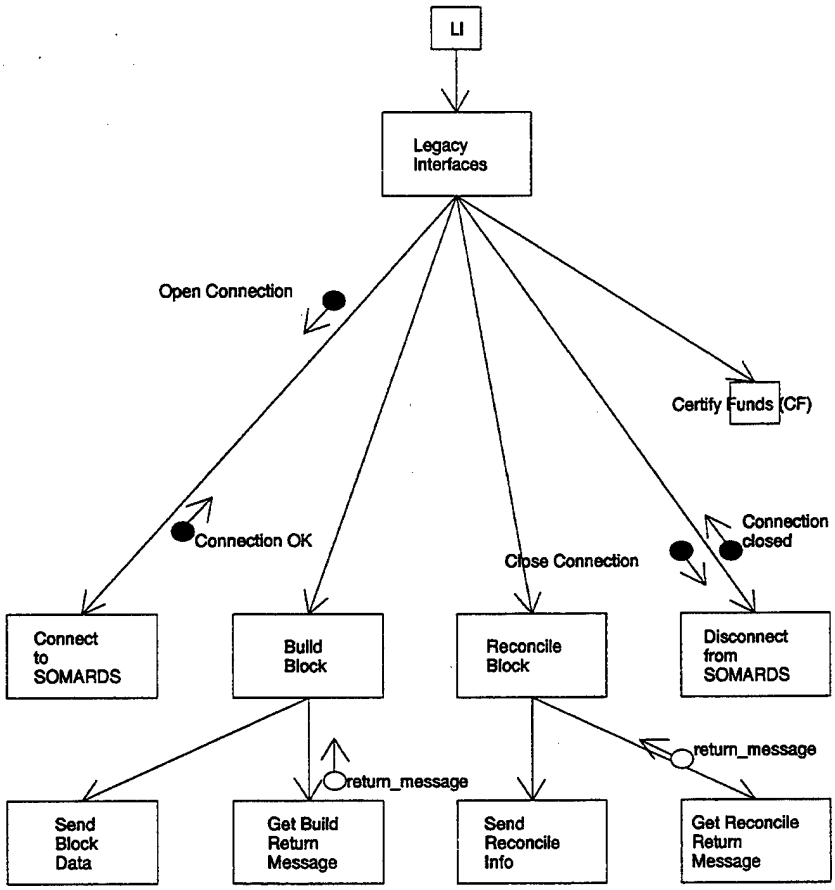


Figure 15. Legacy System Interfaces.

Functional Details:

1. Connect to SOMARDS user interface.
2. Build Block in SOMARDS at start of day.
3. Certify Funds as they become available.
4. Reconcile Block at end of day.
5. Disconnect from SOMARDS interface.

Module: Connect to SOMARDS.

Purpose: Provide an on-line interface to SOMARDS.

Uses: N/A.

Returns: connection_ok.

Functional Details:

1. Telnet to Rock Island IBM mainframe.
2. Proceed to SOMARDS login screen.
3. Login to SOMARDS.
4. If successful, return connection_ok.

Module: Build Block.

Purpose: Build a transaction block for Contract Req to use during the day.

Uses: Send Block Data, Get Build Return Message.

Returns: N/A.

Functional Details:

1. Send Block Data to SOMARDS connection.
2. Get Build Return Message.
3. Repeat steps 1 and 2 until return_message is success.

Module: Send Block Data.

Purpose: Sending transaction block data to SOMARDS for the morning build.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Get today's date and format.
2. Send block to SOMARDS.

Module: Get Build Return Message.

Purpose: Check the status of the build block success.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Receive return_message from SOMARDS connection.
2. Return return_message to parent module.

Module: Reconcile Block.

Purpose: Reconcile the SOMARDS block at the end of the day.

Uses: Send Reconcile Info, Get Reconcile Return Message.

Returns: N/A.

Functional Details:

1. Send Reconcile Info to SOMARDS connection to check balance.
2. Get Reconcile Return Message balance.
3. Check return_message balance against Contract Req total.
4. Send Reconcile Info to SOMARDS connection to adjust balance.
5. Get Reconcile Return Message.

6. Send Reconcile Info to SOMARDS connection to check balance.
7. Get Reconcile Return Message balance.
8. Check return_message balance against zero.

Module: Send Reconcile Info.

Purpose: Send reconcile info to the SOMARDS connection.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Send Reconcile Info to SOMARDS connection.

Module: Get Reconcile Return Message.

Purpose: Receive the balance information from the SOMARDS interface.

Uses: N/A.

Returns: return_message.

Functional Details:

1. Receive return_message from SOMARDS connection.
2. Return return_message to parent module.

Module: Disconnect from SOMARDS.

Purpose: To disconnect from the SOMARDS user interface.

Uses: N/A.

Returns: connection_closed.

Functional Details:

1. Send logout command to SOMARDS connection.
2. Wait for logout complete.
3. Return connection_closed to parent module.

5.9 Certify Funds. Figure 16 shows the "Certify Funds" process structure chart. This module provides a continuous certification process for Contract Req.

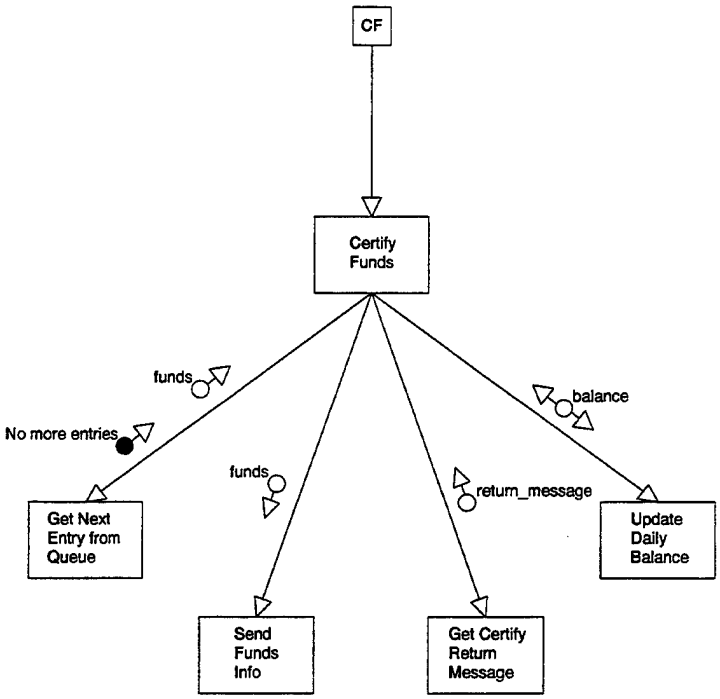


Figure 16. Certify Funds.

The module specifications associated with Figure 16 are listed as follows:

Module: Certify Funds.

Purpose: Continuously certify funds.

Uses: Get Next Entry from Queue, Send Funds Info, Get Certify Return Message, Update Daily Balance.

Returns: N/A.

Functional Details:

1. Get Next Entry from Queue of certifiable contract requests funding information.
2. Send Funds Info to SOMARDS connection.
3. Get Certify Return Message from SOMARDS connection.
4. Put the return_message in the certification return queue.
5. Update Daily Balance if successful.

Module: Get Next Entry from Queue.

Purpose: Retrieve the next contract request funds info.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Scan the certifiable queue for the next entry.
2. Return the funds info to the parent module.

Module: Send Funds Info.

Purpose: Send the funds info to the SOMARDS connection.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Send the funds info to the SOMARDS connection.

Module: Get Certify Return Message.

Purpose: Retrieve the certification results.

Uses: N/A.

Returns: return_message.

Functional Details:

1. Receive return_message from SOMARDS connection.
2. Return return_message to parent module.

Module: Update Daily Balance.

Purpose: To keep a running total of dollars certified.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Calculate new balance using old balance plus the certification amount.
2. Store the new balance.

5.10 Lotus Notes Agent Processes. Figure 17 shows the "Agent Processes" structure chart. This module handles the background process of the legacy system interface queues.

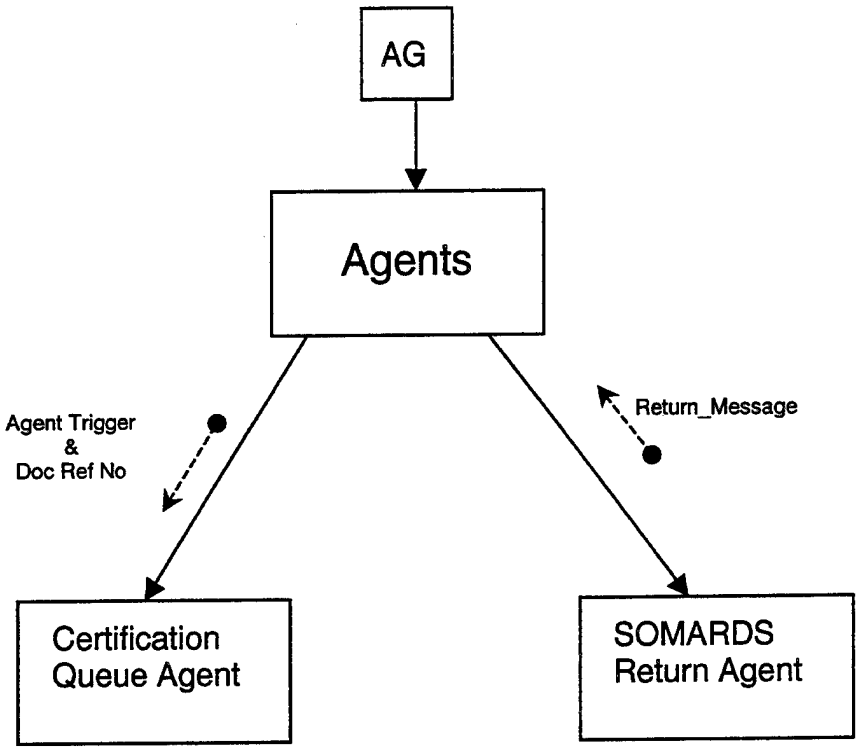


Figure 17. Lotus Notes Agent Processes.

The module specifications associated with Figure 17 are listed as follows:

Module: Agents.

Purpose: To start timed agents and trigger queue agents.

Uses: Certification Queue Agent, SOMARDS Return Agent.

Returns: N/A.

Functional Details:

1. Start timers for SOMARDS Return Agent.
2. Wait for triggers and doc_ref_no from user interface.

Module: Certification Queue Agent.

Purpose: To transfer funding information to the certification queue.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Wait for user interface trigger.
2. Scan for contract request using doc_ref_no.
3. Transfer funding info to certification queue.

Module: SOMARDS Return Agent.

Purpose: To transfer SOMARDS returns from the results queue to the contract request.

Uses: N/A.

Returns: N/A.

Functional Details:

1. Wait for timing trigger.
2. Scan SOMARDS results queue for entry.
3. If certification is "Yes," then mark contract request as certified and set author to special approver.
4. If certification is "No," then set author to requester.

6. References

1. Business Process Reengineering Office. "To-Be Model: Formal Contracts." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, October 1995.

2. McGurin, D. "Contract Req Software Development Plan Version 1: A C-BASS Component." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, to be published.

3. McGurin D. "Contract Req Software Requirements Analysis Version 1: A C-BASS Component." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, to be published.

4. Yourdon, E. *Modern Structured Analysis*. Englewood Cliffs, NJ: Yourdon Press, 1989.

INTENTIONALLY LEFT BLANK.

NO. OF COPIES ORGANIZATION

2 DEFENSE TECHNICAL INFORMATION CENTER
DTIC DDA
8725 JOHN J KINGMAN RD
STE 0944
FT BELVOIR VA 22060-6218

1 HQDA
DAMO FDQ
D SCHMIDT
400 ARMY PENTAGON
WASHINGTON DC 20310-0460

1 OSD
OUSD(A&T)/ODDDR&E(R)
R J TREW
THE PENTAGON
WASHINGTON DC 20301-7100

1 DPTY CG FOR RDE HQ
US ARMY MATERIEL CMD
AMCRD
MG CALDWELL
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 INST FOR ADVNCD TCHNLGY
THE UNIV OF TEXAS AT AUSTIN
PO BOX 202797
AUSTIN TX 78720-2797

1 DARPA
B KASPAR
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

1 NAVAL SURFACE WARFARE CTR
CODE B07 J PENNELLA
17320 DAHLGREN RD
BLDG 1470 RM 1101
DAHLGREN VA 22448-5100

1 US MILITARY ACADEMY
MATH SCI CTR OF EXCELLENCE
DEPT OF MATHEMATICAL SCI
MAJ M D PHILLIPS
THAYER HALL
WEST POINT NY 10996-1786

NO. OF COPIES ORGANIZATION

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL DD
J J ROCCHIO
2800 POWDER MILL RD
ADELPHI MD 20783-1145

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AS (RECORDS MGMT)
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CI LL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

4 DIR USARL
AMSRL CI LP (305)

NO. OF
COPIES ORGANIZATION

8 DIRECTOR
US ARMY RESEARCH LAB
AMSRL IS
J GRILLS
AMSRL CI A
R ROSEN (2 CPS)
AMSRL CS
COL K LOGAN
AMSRL CI E
W JERNIGAN
W MCCOY
B SCHALLHORN
AMSRL SE S
A SINDORIS
2800 POWDER MILL RD
ADELPHI MD 20783-1145

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL EA
P STAY
WSMR NM 88002-5501

ABERDEEN PROVING GROUND

14 DIR USARL
AMSRL CI
N RADHAKRISHNAN
AMSRL CI E
D ULERY (10 CPS)
AMSRL HR SF
B AMREIN
AMSRL WM IM
R MCGEE
AMSRL CS AP
D ORE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1999	Final, January 1998 - March 1998

4. TITLE AND SUBTITLE

Contract Req Detailed Design Report Version 1: A C-BASS Component

5. FUNDING NUMBERS

AC9UA4EC

6. AUTHOR(S)

Denis McGurin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-CI-E
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-MR-459

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document is the third in a series of reports documenting the structured software engineering of Contract Req Version 1.0. "Contract Req Software Development Plan" and "Contract Req Software Requirement Analysis" are the first two titles in the Contract Req technical documentation library. As part of the Corporate Business Application Software System (C-BASS) suite of work flow applications, Contract Req automates preparation of a requestor's procurement package as part of the contracting process for the U.S. Army Research Laboratory (ARL). Three major sections of this report give background for the Contract Req project, explain the structured methodology, and define the design environment. A fourth segment presents a detailed design for Contract Req using the formalisms of structure charts for defining operations and module specifications for defining objects.

14. SUBJECT TERMS

contract preparation, user requirements, software engineering, structured design, structure charts.

15. NUMBER OF PAGES

54

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-MR-459 (McGurin) Date of Report September 1999

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT ADDRESS

_____
Organization

_____
Name E-mail Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)