FUJ00171948

Document Title:	EPOSS Transaction Service - High Level Design
Document Type:	Design
Release:	CSR +
Abstract:	This document provides the High Level Design of the EPOSS Transaction Services
Document Status:	APPROVED
Author & Dept:	Jim Wills – EPOSS Design - Designer
Contributors:	Francesco Chiarini Jim Wills Keith Toh Phil Hemingway
Reviewed By:	Walter Wright Nam Pandher Ian Morrison Lorraine Holt John Pope Hazel Salvat Duncan MacDonald Vin Patel Rex Dixon Nikki O'Sullivan Francesco Chiarini Pete Lindsay Stephan Robson Martin Riddell David Groom Chris Bailey Roger Donato
Comments By:	
Comments To:	
Distribution:	Walter Wright Nam Pandher Ian Morrison Lorraine Holt John Pope Hazal Salvat Duncan MacDonald Vin Patel Rex Dixon Mik Peach Nikki O'Sullivan Francesco Chiarini Pete Lindsay Stephan Robson Martin Riddell David Groom Dai Jones Roger Donato Peter Jeram

0.0 Document Control

0.1 Document History

FUJ00171948

Version No.	Date	Reason for Issue	Associated CP/PinICL No.
0.1	12/5/00	DRAFT Issue	
1.0	27/7/00	For APPROVAL	

0.3 Approval Authorities

Name	Position	Signature	Date
Lorraine Holt	POCL Infrastructure Delivery Manager		
Stephan Robson	Technical Design Authority Manager		

0.4 Associated Documents

Reference	Version	Date	Title	Source
TD/ARC/001	4.6	22/03/00	Technical Environment Description	Pathway
EP/DOC/002	0.1	24/5/00	EPOSS Development – Documentation Roadmap	Pathway
EP/DES/019	0.1	11/04/00	EPOSS High Level Design	Pathway
EP/DES/021	0.1	12/05/00	EPOSS Balancing Service High Level Design	Pathway
EP/DES/020	0.1	12/05/00	EPOSS Reporting Service High Level Design	Pathway
EP/LLD/012	0.1	05/06/00	EPOSS Core Low Level Design	Pathway
EP/LLD/015	0.1	05/06/00	EPOSS Settlement Object Low Level Design	Pathway
EP/IFS/001	0.4	15/06/00	EPOSS/LFS Interface Specification	Pathway
EP/IFS/002	0.1	15/06/00	EPOSS/OBCS Interface Specification	Pathway
EP/IFS/003	0.2	15/06/00	EPOSS/APS Interface Specification	Pathway
BP/FSP/004	4.0	05/03/99	EPOSS Functional Specification	Pathway
EP/DES/002	6.1	7/9/98	Attribute Grammar Catalogue	Pathway
EP/DES/018	0.1	5/4/00	EPOSS Design Specification for CP 2400	Pathway

0.5 Abbreviations/Definitions

FUJ00171948

Abbreviation	Definition
API	Application Interfaces
APS	Automated Payment Service
BES	Benefits Encashment Service
BP	Balancing Period
BT	British Telecomm
CAP	Cash Accounting Period
CP	Change Proposal
CSR	Core System Release
CSR+	Core System Release Plus
DFD	Data Flow Diagram
DLL	Dynamic Link Library
EOD	End Of Day
EPOSS	Electronic Point of Sale Service
HLD	High Level Design
HTML	Hypertext Mark-up Language
LFS	Logistics Feeder Service
LLD	Low Level Design
MIS	Management Information System
OBCS	Order Book Control Service
OPS	Outlet Processing System
PC	Personal Computer
PLU	Product Look Up
POCL	Post Office Counters Limited
Qty	Quantity
RC	Record Count
RDMC	Reference Data Management Centre
SSADM	Structured Systems Design and Analysis Methodology
SV	Sale Value
TED	Technical Environment Description
TIP	Transaction Information Processing
TPS	Transaction Processing System
TV	Television
TXN	Transaction
UCT	Universal Coordinated Time

FUJ00171948

UML	Unified Modelling Language
VAT	Value Added Tax
VB	Visual Basic

0.5 Changes in this Version

Version	Changes
0.1	First Draft
1.0	Changes resulting from review

0.6 Changes Expected

Changes
Further refinement and clarification
Rework as a result of review comments
Changes due to design iterations

FUJ00171948

0.7 Table of Contents

FUJ00171948

FUJ00171948

FUJ00171948

1 Introduction

This document describes the High Level Design of the Transaction Service forming part of the EPOSS In-Day Service. This will be set within the context of the high-level architecture of the EPOSS Application Product.

The purpose of this document is to provide the necessary information for developers to understand the role played by the Transaction Service in order to maintain the system, and also to provide a reference document of the current architecture for future reference. The complete set of EPOSS Application documents and their relationships can be found in EP/DOC/002.

The context of the design is first established within the domain architecture of the EPOSS Product. The major components of the Transaction Service will be identified, their interactions with each other and the rest of the Pathway solution are defined and areas of focus for the high level design will be identified. These areas of focus will subsequently form the basis for the low-level design of the individual product components.

FUJ00171948

2 Scope

The Transaction Service is defined as the service provision made within the Pathway Solution to outlets, which are going to operate Horizon. Specifically, the provision of the Transaction Service forms part of the EPOSS Product.

The High Level Design of EPOSS (see reference EP/DES/019) defines the services provided by EPOSS as forming Domains of the Product's solution. As such the Transaction Service forms a domain of the EPOSS Product, being supported by specific technical architectures defined in the Technical Environment Description (TED), see reference TD/ARC/001.

This document describes the logical view of Transaction Service and is an abstraction of the existing code. The physical view of Transaction Service is defined in code specifications, reference EP/LLD/012 and EP/LLD/015. Each code specification is a further decomposition of the logical view given in this document.

Responsibility for the development and support of the EPOSS Product and hence the applications within the product that provide the Transaction Service, is provided by the EPOSS Development Team within the POCL Infrastructure Delivery Unit of the Pathway Programme.

The purpose of this section is to first establish the scope of the EPOSS Transaction Services High Level Design and to clearly position it within the context of the Technical Environment Description of the Horizon System.

FUJ00171948

2.1 Technical View

The overview of the Horizon application structure and services, similar to that discussed in the Technical Environment Description, is shown in Figure 1. Horizon is the name used by POCL to describe the system made available to Outlet staff. The description of the technical architecture of the ICL Pathway solution for Post Office Counters Ltd at CSR+ and beyond is described by the Technical Environment Description.

The Transaction Service, offered as part of the Pathway solution, is viewed as part of this architecture and remains consistent with its description.

Figure 1 - Overview of Horizon Application Structure and Services

Each product (e.g. APS, EPOSS, OBCS and LFS) at the counter level, is ideally totally independent of the others, and should have no inherent knowledge of the other's implementation.

A major part of Horizon is the Counter PC. Each Counter PC runs under the Windows NT Workstation V4.0 operating system, with specific device drivers to support the Counter peripherals. However, the conventional Windows NT desktop is replaced by the Riposte Login screen and Desktop service, and the Counter Clerk has no access to standard Windows NT facilities.

Each Counter PC runs a set of independent user applications that sit "on top of" Riposte. These applications run within the Riposte environment on the Counter,

FUJ00171948

shown here in Figure 2. The functionality of the EPOSS System comprises three EPOSS Applications, which support the business process of POCL, and the Riposte Desktop provides the skeletal user front end.

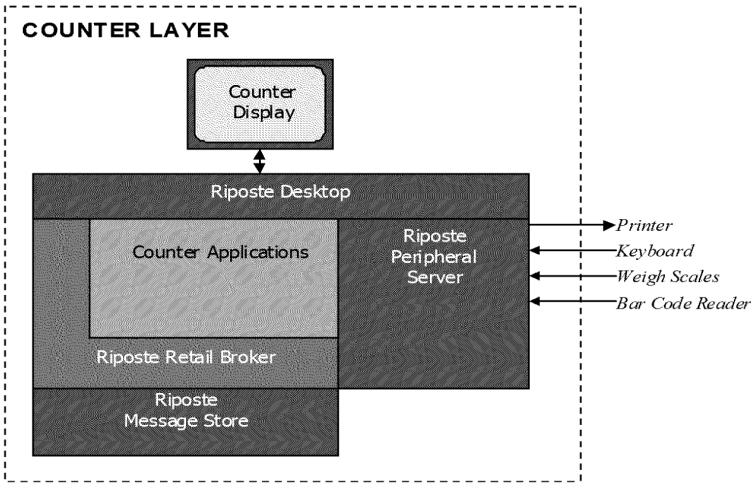


Figure 2 – Environment of Counter Applications

The figure identifies the major components of the Counter Applications Environment, which are:

- Counter Applications
- Riposte Desktop
- Riposte Peripheral Server
- Riposte Retail Broker and
- Riposte Message Store (RMS)

Of these fundamental components, the collection of Counter Applications, is the major focus of this design document as shown in Figure 3. The other components such as the Riposte Desktop,. are supplied by Escher and are, therefore, out of the scope of this document.

FUJ00171948

2.2 Domain View

The EPOSS Transaction Service falls within the bounds of the In-Day-Service and is supported within the architecture of counter applications.

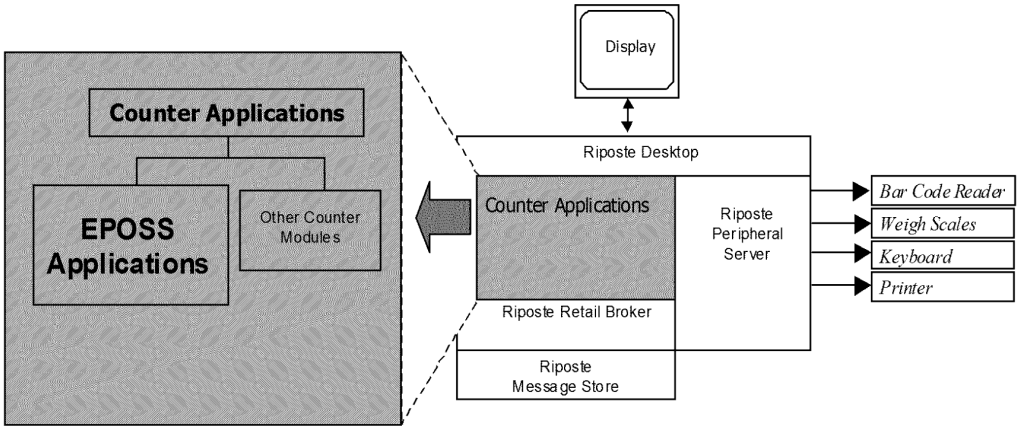


Figure 3 – Domain of Transaction Service is EPOSS Applications

As highlighted in Figure 3, the domain of counter applications comprises software applications, which may be classified as follows:

EPOSS Applications, e.g. EPOSS Core, Settlement Object, Stock Unit, AND MiMan

Other Counter Applications, e.g. APS, OBCS, LFS.

The EPOSS Transaction Service traditionally consists of three EPOSS Application components, Core, Settlement and Scales, shown in Figure 4 below. This document only focuses on the essential components of the Transaction Service, namely Core and Settlement. Scales is outside the scope of this document. The LLD of the Transaction Services is defined in documents EP/DES/012 and EP/DES/015.

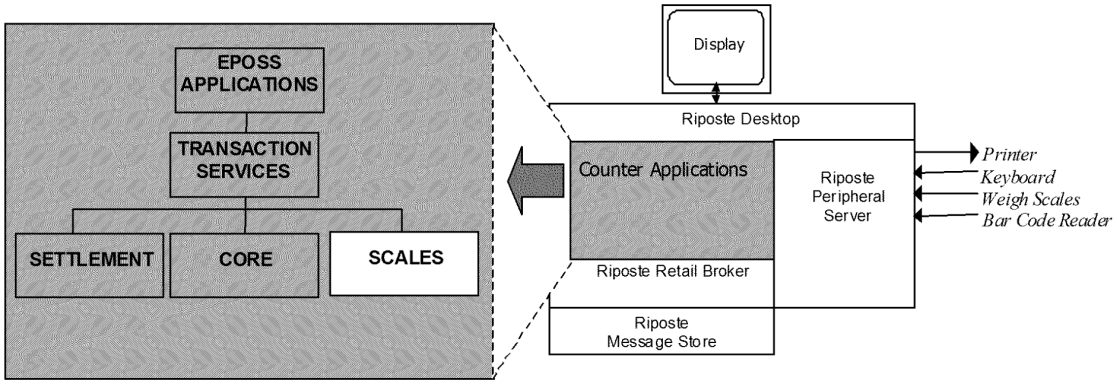


Figure 4 – Physical Components of Transaction Service

FUJ00171948

2.3 Design View

The primary objective of this document is to specify the high level design for the Transaction Service of EPOSS (Figure 5). A secondary objective is to give an overview of EPOSS components and set the context for the documents that address each component. The focus is on the domain of the In-Day-Service, which in turn, may be represented in greater detail by its constituents shown in Figure 6.

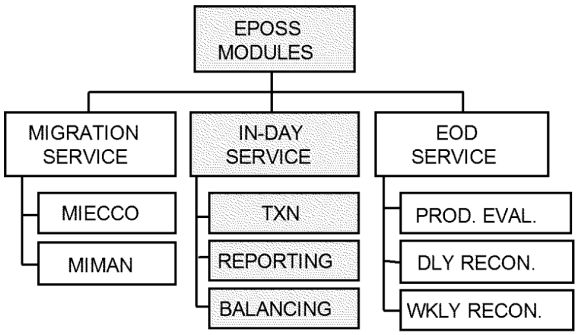


Figure 5 – Transactions are a part of EPOSS In-Day Service

The design documentation of the EPOSS System has been divided into a number of well-defined domains. These domains, which address a number of separate areas of EPOSS Systems functionality, are depicted in Figure 6. These are Transactions, Balancing and Reporting. The scope of this design document is focused on the Transaction processes of the EPOSS System. The processes involved in the Transactions will be identified, showing the interaction with the user and giving an overview of the business rules involved. Balancing and Reporting are described in EP/DES/021 and EP/DES/020, respectively.

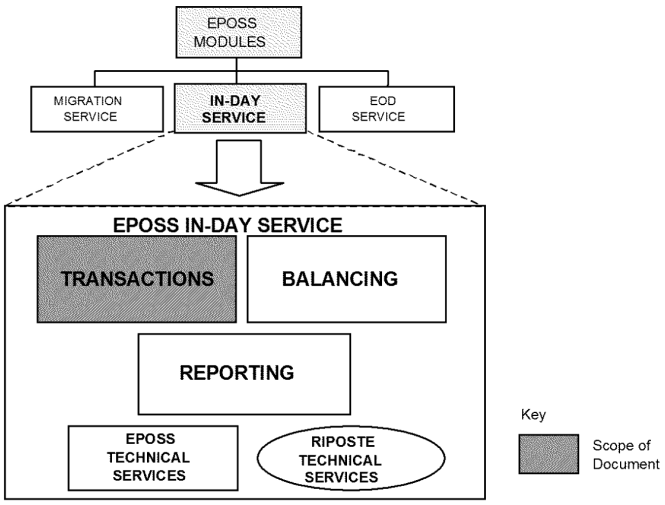


Figure 6 - Scope of Design covers Transactions

FUJ00171948

3 Design Principles

Design and documentation, in general, is carried out at three levels, namely High, Intermediate and Low level. These three levels of design and documentation are depicted in the structure shown in Figure 7. This document contributes to the high level design of the EPOSS Transaction Service.

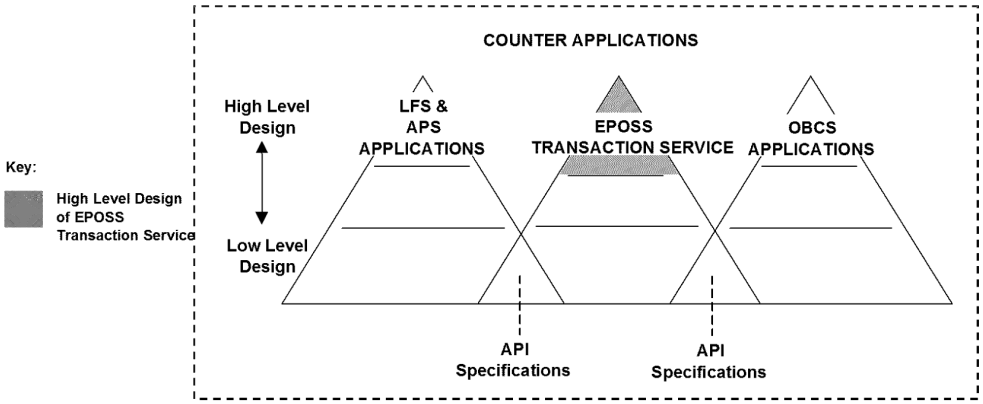


Figure 7 - Structure of Design Documentation

The HLD documentation of the EPOSS Transaction Service is presented in Microsoft Word Version 6.0.

The analysis and documentation for the HLD is facilitated by a number of graphical formalisms and tools for software analysis and design. The high level design is predominantly carried out with the SSADM DFD. Where deemed necessary, UML class and sequence diagrams have been used.

There is a standard set of principles and will apply equally to all development. The principles are:

- Any changes should minimise impact on response times.
- User Dialogue -user interface changes should be consistent with current look and feel and documented during design so that the impact on training and counter procedures can be assessed
- User Dialogue - all user actions should be intuitive and discoverable to minimise counter transaction times and clerk learning curves
- Messages - messages should be consistent with the existing set, be of minimum size consistent with clarity, should allow usage to be obvious and should have unambiguous names
- Architecture - changes should be in line with current architecture and should follow principle of the same actions being executed by the same code wherever these actions are required

Care should be taken to minimise the migration impact of introducing changes to an application in a new release when an earlier version is operational.

FUJ00171948

This document is part of a series of design documents is provided, by the EPOSS Design Team, as a *System Blueprint* which accounts for an existing implementation of the EPOSS Application Product. This articulates the structure of EPOSS as it exists and the System Blueprint includes a number of mandatory Sections including the following:

- Design Principles
- Requirements
- System Components
- Systems Management
- Application Development
- System Qualities
- Solution Implementation Strategy
- Costs, Risks and Time scales

Thus, this design document will identify how the EPOSS Application product fits within the existing architecture (i.e. how it relates to the Technical Environment Description). If changes to the EPOSS Application cannot fit within the existing architecture this will be flagged at an early stage. It does not preclude carrying out the development, but should act as a warning that the architecture may need to be extended and that the development costs cannot accurately be quantified at that stage. Should an architectural enhancement be necessary, this is raised as a *Change Proposal* (CP) to the TED that needs to be implemented prior to the design activities for the development.

FUJ00171948

4 Requirements

The business requirements for the Transaction Service are listed, together with the rest of the EPOSS Product in the EPOSS Home Page **IRRELEVANT**, which provides links to an assortment of EPOSS related information including the customer requirements specifications.

This section describes the entities that are managed by the Transaction Service and outlines the user facilities and services provided for other applications. This provides the technical requirements, which would normally be derived from the business requirements specified for the EPOSS system in BP/FSP/004.

The Transaction Service is primarily responsible for the management of transaction sessions and their individual transactions generated within the OPS environment. A transaction session is a collection of transactions initiated when a product is selected and placed on the Product Stack, via the Desktop Service, on behalf of the user.

During a transaction session the user can return to the home menu, a previous menu or suspend the current session and start another transaction session. The user can also abandon the transaction session and cancel or change the details of one or more transactions on the Product Stack.

A transaction may be presented directly from the Desktop Service or pre-processed by another application before being passed onto the Transaction Service. During a transaction session any number of transactions may take place via any one of the available EPOSS, APS or OBCS application services. A transaction session can also be transferred to another counter.

A transaction may be associated with a product that can be transacted or a service that is provided or a method of payment. A product may be an item sold over the counter such as a first class stamp or an item of stock such as a Milk Token. A service is normally associated with open value commodities such as a British Telecom Bill. Methods of payment, such as cash or vouchers, provide a mechanism for balancing the products and services transacted.

A transaction session is complete when the outstanding balance reaches zero, i.e. the value of products and services required equals the value of payments received. The completion of a transaction session commits each underlying transaction and enables a new transaction session to be initiated.

The Transaction Service must be made aware of the business context in which the user is operating. The user also manages business transactions in one of several modes depending on the type operation required, which involves either handling stock or dealing with customers.

A desktop session is composed of several other session layers in addition to the transaction session. All other sessions, described in the next section, exist to support the transaction session, which is at the lowest level. A desktop session is initiated when the counter system is switched on. A desktop session manages several other sessions described next.

FUJ00171948

4.1 Desktop Sessions

A Desktop Session starts whenever a clerk logs in at a counter. A desktop session is normally terminated when the clerk logs out from the counter before the end of the day. Otherwise an EOD mark for the day is written to the message store and the clerk can continue to enter transactions for the following day without interruption.

A User Session is started when a clerk logs on to the desktop service at a counter and terminates when the clerk logs out. A user session is timed out if the counter is not in use for a predetermined length of time, currently 60 minutes, as defined by Reference Data. A new desktop session can start for the following day anytime after the last EOD mark.

A clerk initiates a Transaction Session from the desktop service when the first sale item is transacted. A menu button associated with a product or service, such as a BT Gas bill, generates an impulse to start an EPOSS transaction. Each product and service has a set of business rules that determine the outcome of the transaction.

A transaction session is normally associated with a specific commodity identifying a product or service. Some commodities are exchanged as multiple units in one transaction but others may be linked to more than one commodity by rules indicating dependency on other products. During a transaction session any existing transaction, displayed on the desktop, can be altered or cancelled (or voided).

A transaction session is complete when a settlement is reached for any number of sale items that have been transacted. Settlement involves one or more payment items, which are only visible to the clerk when a customer is being served. Settlement is automatic for modes involving transactions on items within a stock unit.

Before a transaction session is complete the clerk can return to the home menu or a previous menu to perform other tasks permitted during the session. A transaction session is complete when payment has been initiated and the outstanding balance is zero.

After a transaction has been initiated but before a transaction session is complete the clerk can suspend the current session to initiate a new transaction session. The clerk can swap between the original and new sessions to perform more than one task at a time. No more than two sessions can be open simultaneously.

The current state of a user session is swapped from one counter to another when the clerk logs on to a new counter. If a clerk has a user session open on one counter then logging on to another counter will cause the system to automatically transfer the user session to the new counter. A clerk can only be logged on to one counter at a time so the system will automatically log out the user from the originating counter before transferring the user session to the new counter. A swapped session will continue in the normal manner from its current state at the new counter.

FUJ00171948

4.2 Session Levels

The session layers of a desktop session, described in the previous section, are illustrated in Figure 8 below. A desktop session is initiated when the counter system is switched on. A desktop session manages several other types of session.

A user session starts when the clerk logs on to the Riposte system and terminates as soon as the clerk logs off or after the system has timed out. After performing any necessary balancing from a previous session the clerk is ready to perform business transactions that form part of a daily routine.

The type of operation required for business transactions is established by setting the session mode. Users must decide whether to discharge or replenish items from their assigned stock unit or serve customers at their counter position. Other operations include bulk input (recovery), reversals and housekeeping. Establishing the session mode prepares the system for the type of products and services required for subsequent transaction sessions.

A transaction session can then be started by selecting a product or service from the menu of items provided by the session mode. Several items can be selected including the methods of payment depending on the session mode established earlier.

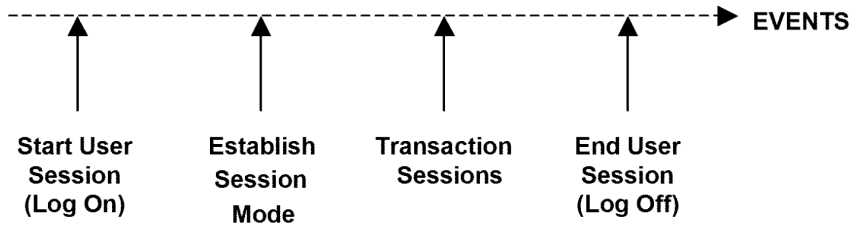


Figure 8 – Context of Transaction Service within a Desktop Session

A more detailed view of the Transaction Session context set within a Desktop Session is shown in Figure 9. The context of a Transaction Session is invoked by Initiate Session and released by Terminate Session, using the session mode.

FUJ00171948

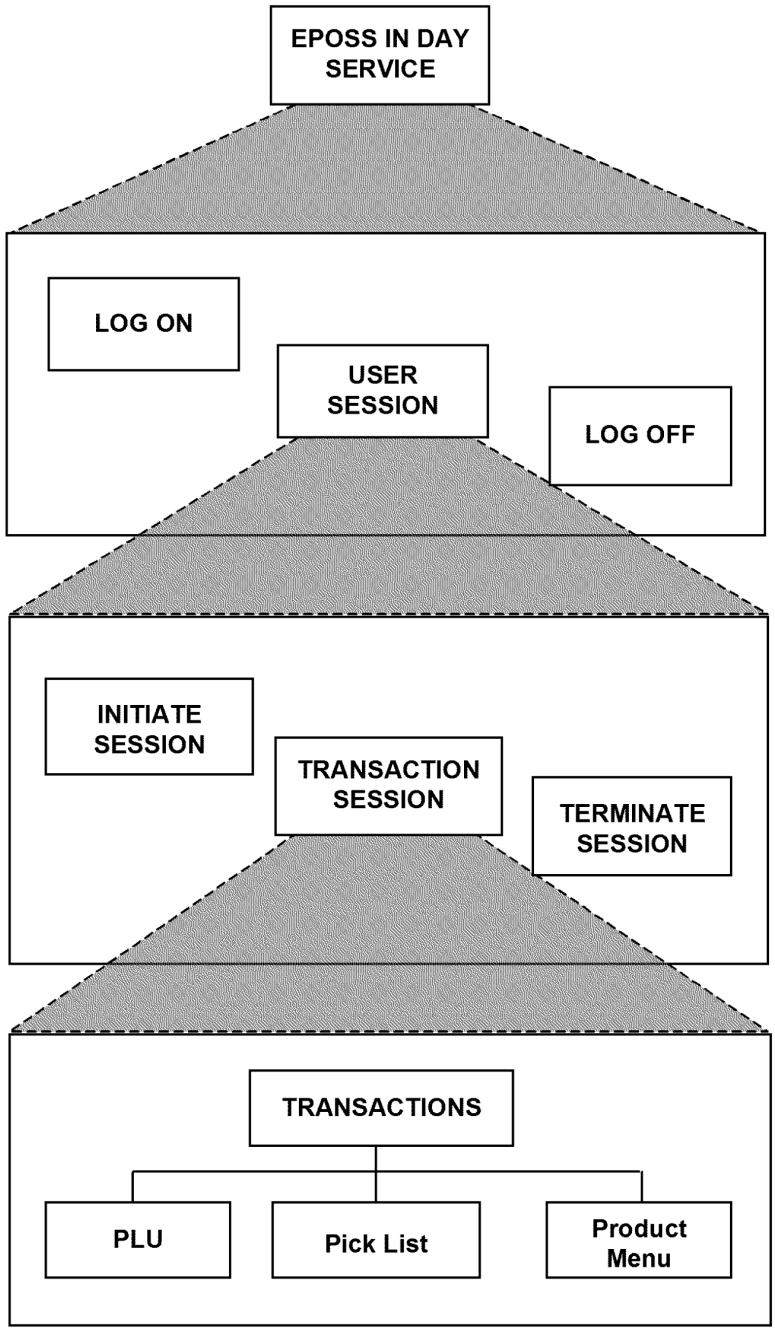


Figure 9 – A Breakdown of the EPOSS Transaction Session

FUJ00171948

4.3 Stock Transactions

After a successful log on to the desktop service, a stock unit will normally be automatically assigned to the user session. One or more users may be attached to the same stock unit depending on the configuration set up by the Post Office manager at an outlet. A Default Stock Unit is allocated when a user has NOT been assigned to a stock unit. Users must check and balance their stock unit levels.

Balancing is an accounting process that enables reconciliation of stock transactions by recording stock and cash levels at convenient times of the day. An opening and closing balance of stock and cash levels enable the declaration of any losses or gains due to discrepancies between the two.

When a stock unit is replenished with merchandise from another stock unit within the same outlet a transfer in operation has taken place. The depletion of merchandise from the other stock unit within the same outlet is called a transfer out operation. More than one stock unit, excluding the default stock unit, must be allocated to an outlet before a stock transfer operation can take place.

When a stock unit is replenished with merchandise received from the distribution centre or another outlet a remittance in operation has taken place. The depletion of merchandise from a stock unit for transfer to another outlet is called a remittance out operation. A remittance can be made on behalf of the Auto-Distribution Centre, a Post Office Client, another Post Office outlet or a Supply Division.

Any commodity contained within a stock unit is subject to revaluation, i.e. a change in its stock value. The change in stock value of a single commodity, such as a stamp, will also affect the stock value of the same commodity sold in multiple units, such as a book of stamps.

An existing transaction can be reversed after settlement depending on the product transacted. A reversal may have to take place on the same day as the original transaction (i.e. prior to end of day) or may be rejected if the counter position being used is not communicating with other counters in the outlet.

A linked reversal is used to reverse a transaction that is within the same balancing period and Cash Accounting Period as the original transaction. An unlinked reversal is used when no receipt is available for the original transaction or to perform corrections for balancing the accounting system. Although a reversal may be linked to an existing transaction the payment associated with the reversal cannot be linked.

A single transaction, settled during a remittance session, can be reversed using the transaction identifier printed on a receipt. If a receipt is not available then the user can reverse a transaction by transacting the product with a negated sale value.

A transfer in session and its individual transactions cannot be reversed. Individual transactions within a transfer out session are also irreversible. A transfer out session can be reversed. Transfer in sessions can only be reversed via a transfer out session.

FUJ00171948

4.4 Customer Transactions

After balancing stock levels or supplying the overnight cash holding or at the start of working week the clerk can start serving customers from a given counter position. A customer session is initiated from the desktop via the Serve Customer menu item, or by reading a token as described in the next section.

A Customer Session is a transaction session designated for transacting one or more products and/or services over the counter. Before completion of a customer session a transaction can be cancelled or altered as described for other transaction sessions.

The settlement phase of a customer session is initiated as soon as the first payment item is selected. An outstanding balance is the total sum of the sale items minus the sum of the payment items. Settlement is achieved when an outstanding balance of zero is reached.

During a customer session any number of customer transactions can take place for any one of the applications available within OPS. APS and OBCS transactions are initiated by the use of a token device such as a bar-coded document, magnetic card or smart card. Other transactions are initiated from the desktop service by selecting a menu button associated with a product or service, such as a BT Gas bill.

A customer session is complete when the last payment received from the customer achieves a zero session balance. Payments can be made from any one of the Methods of Payment available for the products and services transacted. Each session must check the validity of any chosen method of payment in exchange for the items purchased by the customer.

Bulk Input is a method used to generate customer transactions that have been recorded manually. This is a fast method of data input that enables recovery from a system failure at an outlet.

A customer can also request a refund or reversal of a customer transaction after settlement of a customer session. A refund involves the repayment of money to the customer for goods previously sold at an outlet. A reversal enables a previously paid bill to be annulled.

A single transaction, settled during a customer or bulk input session, can be reversed using the transaction identifier printed on a receipt. . If a receipt is not available then the user can reverse a transaction by transacting the product with a negated sale value so that a refund can be given.

FUJ00171948

4.5 Office Transactions

Housekeeping is deployed to correct errors made during one or more transaction sessions or to apply adjustment transactions as required for the office balance process. Housekeeping transactions involve the movement unsettled cash, unpaid cheques and unpaid vouchers to and from a special office level suspense account.

Non-Accounting Data is used to establish quantities of a particular type of service transacted such as fishing licenses. POCL need this information to establish and forecast inventory levels. This facility is also used to record volume of business for designated clients, such as Royal Mail. All products and services transacted in this mode have zero value.

Parcel Traffic is deployed to determine volumes and values of parcels, already transacted during a customer session, serviced by Parcel Force as opposed to Royal Mail. Parcels are normally transacted during a customer session using the Scales Service.

FUJ00171948

4.6 Reference Data

4.6.1 Transaction Entities

Transaction Service is driven by Reference Data collections. For each transaction, Reference Data is extracted by the Transaction Service to establish the attributes and properties of the product or service involved. Reference Data may also dictate the capture of additional data for a given product or service during a transaction session.

Transaction Service also has to take account of the session mode during the transaction of a sale or payment item. The validity of a transaction and its results is dependent on the session mode and the associated attributes and properties of the product or service involved contained in product collections defined by Reference Data.

Product collections, held as Reference Data, identify the business objects and define the business rules for their transaction. Transaction Service utilises the given business objects and interprets the business rules to ensure they are implemented for each transaction generated by the system.

A standard set of products and services are specified for all Post Office outlets but some can provide special commodities. Product collections are used to configure each Post office outlet to establish the set of commodities that they can transact. Each product collection used by the Transaction Service is defined in the following sections.

Individual transactions may be initiated directly from a menu button or token device or indirectly via selecting a product from a pick list or entering a product number on the PLU No panel. A PLU List panel and a Product Group panel enables the selection of core products and special products, respectively.

All core products can be identified on a button, known as a product button, within the menu hierarchy of the EPOSS system. Menu buttons for non-core products will also appear only if the product is sellable at the given outlet. Buttons contain a caption containing the product name, a caption identifying the keystroke on the keyboard and a graphical icon defined by Reference Data.

Some products sold over the counter are allocated a PLU Number in the Reference Data supplied by POCL. These products can be transacted using the PLU No panel or viewed and transacted using the PLU List panel. In both cases Reference Data is used to construct a list of products that can be transacted for each session mode.

FUJ00171948

4.6.2 Transaction Rules

Reference Data can define items ranging from a simple fixed price product, such as a stamp, to complex open value services like BT Bills. Items may be defined in reference data as fixed price, default price or open price. This establishes the rules for calculating the value of items transacted.

If a fixed price item is sold no further interaction with the clerk is required and the transaction total is calculated from Quantity * Unit Price = Sale Price. An icon representing the transaction and displaying the quantity and sale price is displayed on the Product Stack. The total value of transactions on the Product Stack, displayed on the Finish button, is incremented in line with the Sale Price calculated.

If an item is designated as a default price the Transaction Service will display a price panel with the default price, during its transaction, and prompt the clerk to accept or alter the displayed price. The transaction total is calculated from Quantity * Unit Price = Sale Price. An icon representing the transaction and displaying the quantity and sale price is displayed on the Product Stack. The total value of transactions on the Product Stack, displayed on the Finish button, is incremented in line with the Sale Price calculated.

If an item is designated as open price the Transaction Service will display a price panel with no value, during its transaction, and prompt the clerk to enter the price. The transaction total is calculated from Quantity * Unit Price = Sale Price. An icon representing the transaction and displaying the quantity and sale price is displayed on the Product Stack. The total value of transactions on the Product Stack, displayed on the Finish button, is incremented in line with the Sale Price calculated.

A user may input the required quantity prior to selection of a product. Reference Data may impose a minimum or maximum quantity on a given product or service. If such an item is transacted and the quantity is outside the permitted range Transaction Service will display a quantity panel prompting the user to alter the number of items being transacted. A quantity may also be determined for a given item from a given amount of money using the Shopping panel. This will result in rejection if the quantity is outside the permitted range.

Reference Data may impose a minimum or maximum price on a given product or service. If such an item is transacted and the total transaction value is outside the permitted range Transaction Service will display a price panel prompting the user to alter the price of the item(s) being transacted.

In addition to the constraints imposed by Reference Data there are implicit rules imposed by the back-end applications. For instance the maximum quantity supported by TPS is 5 digits (on any product under any mode) so it should not be possible to transact quantities greater than 99999. Similarly it should not be possible to transact a value greater than £9,999,999.99. When a sale item is added to the product stack and the maximum product stack value is exceeded then a system error will be output and the sale rejected.

FUJ00171948

4.6.3 Transaction Attributes

Attributes of items that can be transacted and the rules governing their transaction are given in product collections of reference data. A table summarising the main features is given below and further details are defined in the appendix.

Name	Description
Product Number	Numeric value which defines a product code uniquely identifying the item
Product Name	A short, medium and long name
Retail Price	Sale value of a single item (including VAT if applicable)
Single Quantity Value	Sale value that item must be divisible by when sold in multiple quantities, e.g. Lottery Instants must be transacted in multiples of £1
Price Range	Minimum and maximum sale values of open priced items
Supply Range	Minimum and maximum quantity of items that can be transacted in multiple units
Instruction Script	Instructions to the clerk displayed during the transaction of item
Number of Receipts	The number of receipts to be produced at the end of a transaction session
VAT Identity	Code identifying VAT Code
Owner Identity	Code identifying the owner of item, i.e. POCL or one of its clients
Price Override	Indicates whether the retail price can be overwritten by the user
Transaction Reversal	Indicates whether the transaction of the item can be reversed after committing to message store
Inventory Tracking	Indicates whether the inventory of the item must be tracked continuously
Transaction Refund	Indicates whether a refund can be given after a transaction is committed to message store
Void Transaction	Indicates whether a transaction can be cancelled during a transaction session
Core Item	Indicates whether the item is one of the core products or services offered by all Post Office outlets
Change Sign	Indicates whether the retail value of the item can change sign, i.e. can act as either a sale item or a payment item

FUJ00171948

4.6.4 Product Collections

Product collections and their relationships within reference data, illustrated in Figure 10, are read and processed by Transaction Service. During initialisation or a transaction session reference data is transformed into objects that can be manipulated more easily. For example, transaction and product objects are created for every transaction invoked.

EPOSS Products is composed of all commodities that can be transacted at a given Post Office counter. The Method of Payments for products exchanged over a counter is also defined in EPOSS Products. All EPOSS, APS and OBCS products and payments are declared in EPOSS Products. StartSale and EndSale attributes establish the times when a given commodity can be transacted at an outlet.

EPOSS Tokens are used to identify APS Clients or OBCS Beneficiaries and their associated commodities. Token Impulses identify, transform and route data from input devices to initiate APS or OBCS transactions for processing by EPOSS. Token Impulses are outside the scope of this document.

An EPOSS product can be transacted or exchanged in several modes defined by Product Modes. A customer transaction can be cancelled or suspended during a customer session or refunded or reversed after a transaction has been agreed and committed. A transfer, remittance and revaluation of commodities associated with a stock unit also affect the modes of a product transaction.

EPOSS product codes and their modes, known as PLU Impulses, are used by a clerk to access specific products by its product identity. PLU Impulses are also used to view the list of products held at an outlet. Only a subset of products can be referenced in this manner by their product identity.

An EPOSS product may belong to a special category of products or services defined by Product Groups. A product group is a collection of products and services that enable the configuration of a pick list during a customer session. Product groups such as Travel Schemes are represented in the Local Schemes menu, displayed from the Serve Customer menu.

Tilda EPOSS Products denote the set of non-core EPOSS products that are available to specific outlets. An attribute, named Depend, in Tilda EPOSS Products indicates whether a non-core product is applicable to a given outlet.

Desktop Buttons contains an icon for each commodity that is represented as a menu button on the desktop for initiating a transaction. When a commodity is transacted the icon displayed on the Product Stack is derived from Product Stack.

Mode Parameters defines the menu of products and services that are displayed when the clerk selects a session mode such as Serve Customer. Automatic settlement of a transaction is determined by the existence of the attribute SettlementProduct. After settlement the attribute AlwaysPrintReceipt controls whether a receipt is automatically printed.

Cash Account Mappings determine the entry for each customer transaction in a Cash Account Report. EPPOS Nodes contains the structure for placing transactions into a Cash Account Report. See EPDES020 for a more detailed description of Cash Accounts.

FUJ00171948

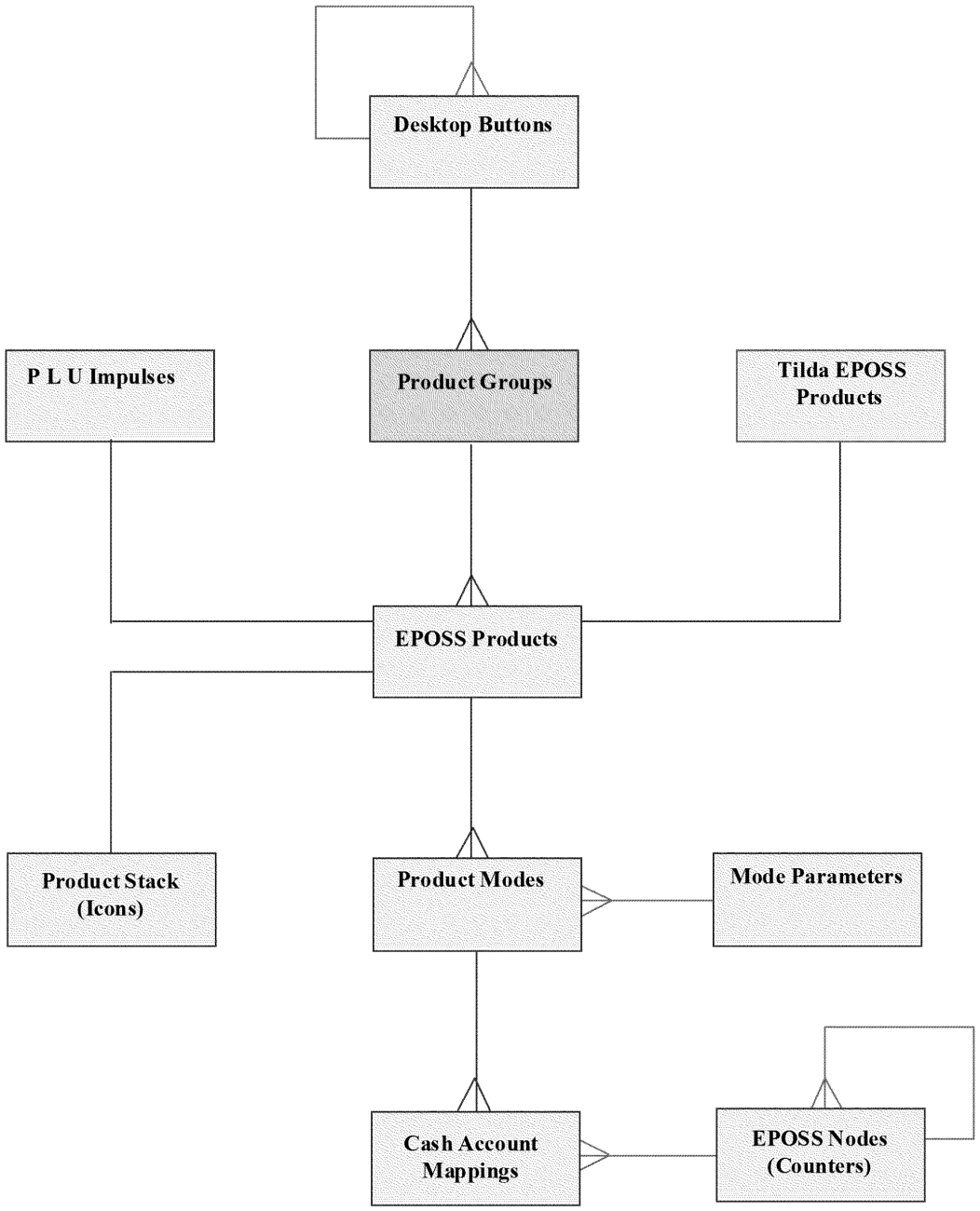


Figure 10 – Reference Data defining Product Collections at each Outlet

FUJ00171948

4.6.5 EPOSS Products

EPOSS Products include POCL products and services such as First Class Stamps, Postal Orders, TV Licences, British Telecom Bill or DSS Order Books for OBCS transactions or APS Client commodities such as an Electricity Bill or Water Bill. EPOSS Products also includes settlement products such as Cash and Cheque to enable the payment of products and services transacted. There are several methods of payment available to customers served at a counter position.

Each record in EPOSS Products has a product number identifying the product and a set of attributes such as fixed price, minimum sale value, maximum sale value, minimum quantity and maximum quantity. Other attributes define dependencies on other products and additional information that must be supplied by the user.

The Session Effect (SE) attribute describes the effect of the sale value of a product or service on the session balance when the item is transacted. A value of 'In' or 'Out' will increase or decrease the balance due to the Post Office, respectively. The value of this attribute determines whether a product or service is a sale item or payment item. A payment item, also known as a settlement product, decreases the balance. The Adopt Settlement Sense (AS) attribute allows a product or service to switch between a sale and payment item when its value is TRUE.

An item in EPOSS Products may be transacted in several session modes specified by POCL in Reference Data. A session mode can be categorised into one of three types of business transaction, namely customer, stock or accounting. Each type of business transaction is co-ordinated separately by Transaction Service and different sets of rules are applied for transacting each item within a transaction session.

Customer transactions are serviced in Serve Customer, Bulk Input (Recovery) or Reversal mode. A movement of stock can be transacted in transfer, remittance and revaluation modes. Housekeeping, Non Accounting Data and Parcel Traffic modes are office transactions for correcting errors, managing inventory items and recording volumes.

An overview of attributes and how they are processed for customer and stock transactions is given below. There are no special attributes for office transactions as the business rules are similar to customer transactions. Attributes for stock transactions are subdivided into value and non-value products. The complete set of attributes for EPOSS Products is given in the appendix.

FUJ00171948

4.6.5.1 Customer Data

A transaction can only be cancelled within a customer session that has not been committed to the message store providing the VO is TRUE. The user must first select the Bin button on the screen and then select the transaction on the Product Stack to be voided. Void transactions are not committed to the message store.

Once a transaction that cannot be voided is added to the Product Stack cancellation is prevented. The transaction must be committed to message store but can be reversed if RV is TRUE. Reversal creates a compensating transaction to effectively annul the original transaction.

A customer session can only be abandoned before a settlement is reached if all the transactions on the Product Stack can be voided. Transactions already settled and committed cannot be voided although they can be reversed in a separate reversal session, if the reference data supplied by POCL enables reversal of the product.

When FP is TRUE, the Sale Value of a selected product is fixed, at a given retail price RP, and cannot be altered by the desktop service. Otherwise the clerk is presented with a dialog box, supplied by the desktop service, for entering a sale value that must lie between MnV and MxV inclusive. Multiple units defined by the Quantity control on the desktop service for a selected product must also lie between MnQ and MxQ inclusive.

A product may be associated with one or more Preconditions given by the attributes PreCondition, PCProdNo and ProductNo. If a precondition exists then a customer transaction cannot take place unless the products identified by the precondition have already been transacted as part of the customer session.

A product may also be associated with Additional Data, used to prompt the user for the appropriate additional entries prior to evaluating or prompting the user for the sale price. Additional data defines the order, name and description of each additional field of information. A message prompt for each additional field of information is also supplied for display on the screen.

Additional Data also provides the format, type and length of additional fields of information. Other characteristics include positioning of their screen content for display on a panel. Minimum and maximum permitted values, default values and rules for updating new values are also specified. Decimal places and leading zeros can be defined for numerical data or currency values.

A product that has already been committed during a customer session may be reversed or refunded when RV and RF are TRUE, respectively. Otherwise any attempt to perform these operations must be ignored or rejected.

FUJ00171948

4.6.5.2 Stock Data

4.6.5.2.1 Value Products

Stock transactions are dependent on the session mode given in the Product Modes collection. A session mode is based on the type of operation required by the user, which involves transfers, remittances and revaluation of items of stock. The movement of cash and stock items can be classed as internal transfers or external transfers. This equates directly to transfer and remittance session modes, respectively.

A transfer out operation involves identifying the receiving stock unit from a pick list of stock units assigned to the outlet. The source and destination stock units must be operating in the same Cash Account Period defined by Reference Data. A menu of products is displayed containing only those items of cash and stock identified in the POCL Reference Data as value stock.

The user will transact the products, quantities and sale value of items to be transferred similarly to a customer session. The user may complete or abandon the session by voiding each transaction individually. The session is automatically settled against a settlement product and a session receipt is always printed on completion.

A transfer in operation involves selecting an item from a pick list of pending stock units transferred out to the stock unit assigned to the current user. The originating stock unit, transfer session identifier and total value identifies each stock unit transferred out.

Each stock unit transferred in and committed to the message store will generate the same transactions originally generated by the transfer out operation. The sale value of each item will be negated to balance stock transfer operations. The session is automatically settled against a settlement product and a session receipt is always printed on completion.

A remittance out operation involves identifying the external source to which the selected items are to be sent. This is determined from the session mode, which may be one of Auto Distribution Centre, Client or Other Post Office. Similarly, for the remittance in operation.

The user will transact the products, quantities and sale value of items to be remitted similarly to a customer session. The user may complete or abandon the session by voiding each transaction individually. The session is automatically settled against a settlement product and a session receipt is always printed on completion.

FUJ00171948

4.6.5.2.2 Non-Value Products

During the declaration or confirmation of non-value stock, processed by LFS, there are some products with a sale value that must be transacted with a zero value. Products in this category are items such as Travel Tickets, Milk Tokens, Utility Tokens. See EP/DES/018.

POCL need to remit non-value stock to or from the Auto-Distribution Centre at zero value even though some products have a fixed value. These products must still continue to be transacted at a fixed value specified by reference data in all other modes.

EPOSS must change the sale value of a product to zero in ROAD and RIAD modes when all the following conditions are satisfied.

Logistics Accounting Item (LACCI) is False

Logistics Inventory Item (LINVI) is True

Value Stock (I) is False

Not a member of an Inventory Item Structure (No ACCI attribute)

FUJ00171948

4.6.6 Product Modes

A product may be transacted in one of several states dictated to by the **M** attribute in **Product Modes**. The states that can be in effect for a given product are given by session modes in the table below. Session modes given in italic are being withdrawn by Change Proposal 2570.

EPOSS must ensure that a product is always in a valid state by checking the prompts from the desktop service. An EPOSS commodity can only be transacted in the modes defined by the set of **M** attributes given in **Product Modes** or **PLU Impulses**. The complete set of attributes for Product Modes is given in the appendix.

Session Mode	Session Name
SC	Serve Customer
TI	Transfer In
TO	Transfer Out
RIAD	Remittance In for Auto-Distribution
ROAD	Remittance Out for Auto-Distribution
RICL	Remittance In for Client
ROCL	Remittance Out for Client
RIOP	*Remittance In for Other Post Office*
ROOP	*Remittance Out for Other Post Office*
RISD	*Remittance In for Supply Division*
ROSD	*Remittance Out for Supply Division*
RODC	Remittance Out for Data Processing Centre
ER	Existing Reversal
RV	New Reversal
RU	Revaluation Up
RD	Revaluation Down
NAD	Non Accounting Data
HK	Housekeeping
PT	Parcel Traffic
REC	Recovery (Bulk Input)
DDP	Declare Discrepancy – Positive
DDN	Declare Discrepancy – Negative
SAP	Stock Adjustment – Positive
SAN	Stock Adjustment – Negative

FUJ00171948

4.6.7 PLU Impulses

PLU Impulses can be used either to generate a pick list for looking up a product or access a specific product by its product identity. Operational modes, also defined by the M attribute in Product Modes, are replicated to improve user response times. The complete set of attributes for PLU Impulses is given in the appendix.

4.6.8 Product Groups

EPOSS products that belong to a special category of products or services are defined in a collection named **Product Groups**. Product Groups contains a group name field defining the category of each product or service. The complete set of attributes for Product Groups is given in the appendix.

4.6.9 Desktop Buttons

Desktop Buttons contains the bit map for the icon presented on menu buttons of the desktop service. An icon is used to display a product or service on one of the menus displayed on the desktop during a transaction session. The complete set of attributes for Desktop Buttons is given in the appendix.

4.6.10 Tilda EPOSS Products

Tilda EPOSS Products denote the set of non-core EPOSS products that are available to specific outlets. An attribute, known as the depend flag, in EPOSS Products indicates whether a non-core product is applicable to a given outlet. The complete set of attributes for Tilda EPOSS Products is given in the appendix.

4.6.11 Product Stack

When a transaction is about to be committed, the icon of the associated product must be derived from the **Product Stack**. This icon is used to display a product on the Product Stack to represent one of the transactions that has taken place during a transaction session. This ensures a product is always presented in a consistent manner. The complete set of attributes for Product Stack is given in the appendix.

FUJ00171948

4.7 Transaction Data

Transaction Service is responsible for managing all business transactions and creating transaction records in the message store area. The transaction record generated is based on product collections held in reference data of the message store. This information provides the business rules for transacting all items.

The product number of the sale or payment item selected is used to look up its details within product collections of reference data. All the necessary information required to transact the item are attached to the transaction.

A transaction is initiated when a sale or payment item is selected from a Transactions menu such as Serve Customer or when input is received from another peripheral device such as a bar code reader. Some items have mandatory links with another item, which may result in several transaction records being generated.

A product, service or payment is transacted during a transaction session, via Retail Broker, when the item appears on the Product Stack. A transaction on the Product Stack may consist of several records that constitute a single transaction. For example, if a postal order is transacted, its associated fee generates a second record.

A transaction session may consist of the sale of a single product or multiple products or services that terminate when settlement is reached with one or more payment items. At this point all the transactions on the Product Stack are committed and transaction records generated are written to the message store area.

When the outstanding balance is zero the Transaction Service passes a request to the Retail Broker to commit all transaction records, for the current transaction session, into message store. The transaction records of the current session are deleted and items on the Product Stack are cleared ready for the next transaction session.

The data model for transaction data consists of three levels of information, in attribute grammar format, generated individually by the Transaction Service, Retail Broker and Riposte Service, respectively. Transaction Service provides details of the item(s) transacted at the innermost level. Riposte Service and Retail Broker supplies additional header and trailer information at the outer levels.

Attribute grammar is a string of characters delimited by < and > characters (angle brackets). Each data item within the string is also delimited by < and > characters. A data item starts with a defined attribute or tag followed by a colon and a value or NULL. An example of a transaction record in attribute grammar format showing the information generated by the Transaction Service, Retail Broker and Riposte Service is given in the following subsections. There are two examples illustrating the attribute grammar strings for the sale and settlement of a first class stamp.

Other products may be linked to another product, identified by Precondition attributes, or require additional information given by attributes within Additional Data. Other types of transaction such as existing reversals have special attributes such as OMode and OModeV giving the original session mode and its version number, respectively. See EP/DES/002.

FUJ00171948

4.7.1 Attribute Grammar for a Transaction

The attribute grammar generated for the sale of a first class stamp during a customer session is given below. The Riposte Service supplies a header and a trailer, shown in italic, to data supplied by Retail Broker. Retail Broker provides a header, shown in the normal font, to data supplied by Transaction Service, which is shown in bold.

The attribute grammar generated for the payment of the stamp during the same customer session is shown in the next subsection. The sale of an item must have one or more associated payment items to achieve settlement before being committed to message store. A full definition of attribute grammar is given in EP/DES/002.

<Message:

<GroupId:123456><Id:1><Num:59301><Date:14-Jun-2000><Time:13:33:32>
<User:MIGR01> <Expiry:35><TranStartNum:59301>

<TxnData:

<SessionId:44-123456-1-59300-1><TxnId:44-123456-1-59300-2><Container:SU1>

<Start:<Date:14-Jun-2000><Time:13:33:28><TF:2>>

<End:<Date:14-Jun-2000><Time:13:33:28><TF:3>>

<Mode:SC>>

<Application:EPOSSAppMain><EPOSSTransaction:

<ProductNo:19><PVer:11><Qty:1><SaleValue:0.26>

<INVI:<ProductNo:19><PVer:11><Qty:1>>

<BlackBoxData:<M:SC><V:12><S:1>>

<TranType:S><PM:<L1:1704><L2:2055><L3:3007><L4:3008><L5:3017>>

<SM:>>

<Credit:26>

<CRC:325DA6C0>>

FUJ00171948

4.7.2 Attribute Grammar for a Settlement

The attribute grammar generated for the payment of a first class stamp during a customer session is given below. The Riposte Service appends a header and a trailer, shown in italic, to data supplied by Retail Broker. Retail Broker provides a header, written in normal font, to data supplied by Transaction Service, which is shown in bold.

The attribute grammar generated for the sale of the stamp during the same customer session is shown the previous subsection. A payment item can only be selected when there are one or more sale items on the Product Stack. One or more payment items may be transacted to achieve settlement and committing to message store.

<Message:

<GroupId:123456><Id:1><Num:59302><Date:14-Jun-2000><Time:13:33:32>
<User:MIGR01>

<Expiry:35><TranStartNum:59301>

<TxnData:

<SessionId:44-123456-1-59300-1><TxnId:44-123456-1-59300-3><Container:SU1>

<Start:<Date:14-Jun-2000><Time:13:33:32><TF:3>>

<End:<Date:14-Jun-2000><Time:13:33:32><TF:4>>

<Mode:SC>>

<Application:EPOSSAppMain><EPOSSTransaction:

<ProductNo:1><PVer:13><Qty:-1><SaleValue:-0.26>

<BlackBoxData:<M:SC><V:15><UnitPrice:0.26><S:1>>

<TranType:S><PM:<L1:1001><L2:1000><L3:3003><L4:3008><L5:3017>>

<SM:>>

<Debit:26>

<CRC:1272AB9E>>

FUJ00171948

5 System Components

5.1 Logical Architecture

The High Level Logical Design of the Counter Applications, showing the EPOSS-In-Day Service is shown in Figure 11. This shows the context of EPOSS Transaction Service in relation to the overall Pathway solution.

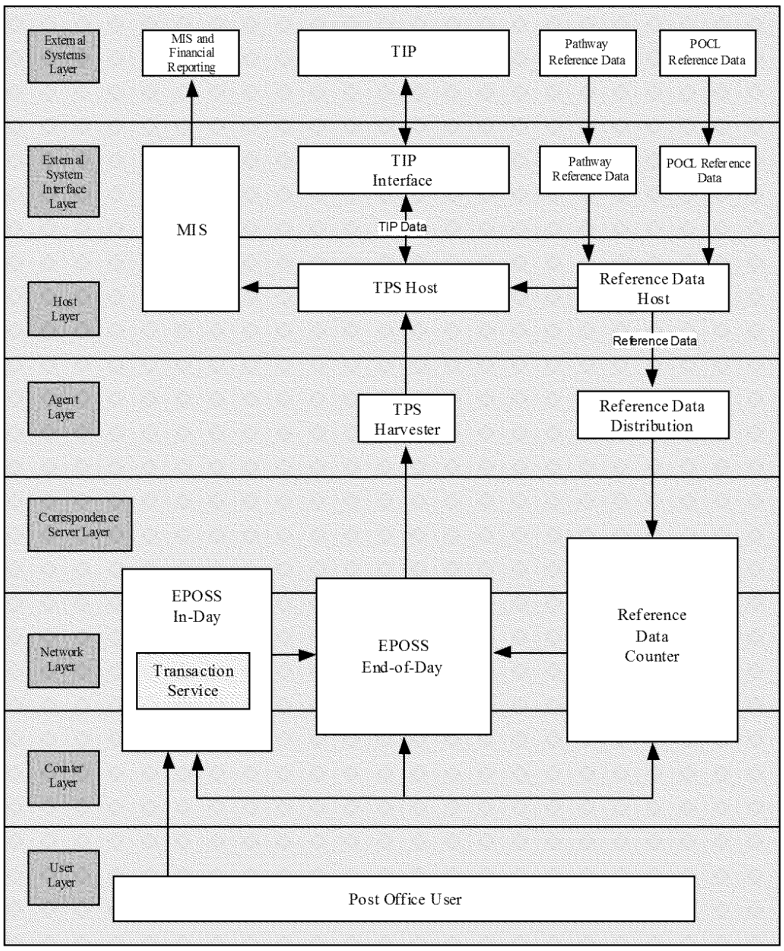


Figure 11 – High Level Design of Counter Applications showing EPOSS Transaction Service

FUJ00171948

5.2 Physical Decomposition

The architecture of the EPOSS System, focusing on the Desktop, Technical Services and EPOSS Applications is shown below in Figure 12.

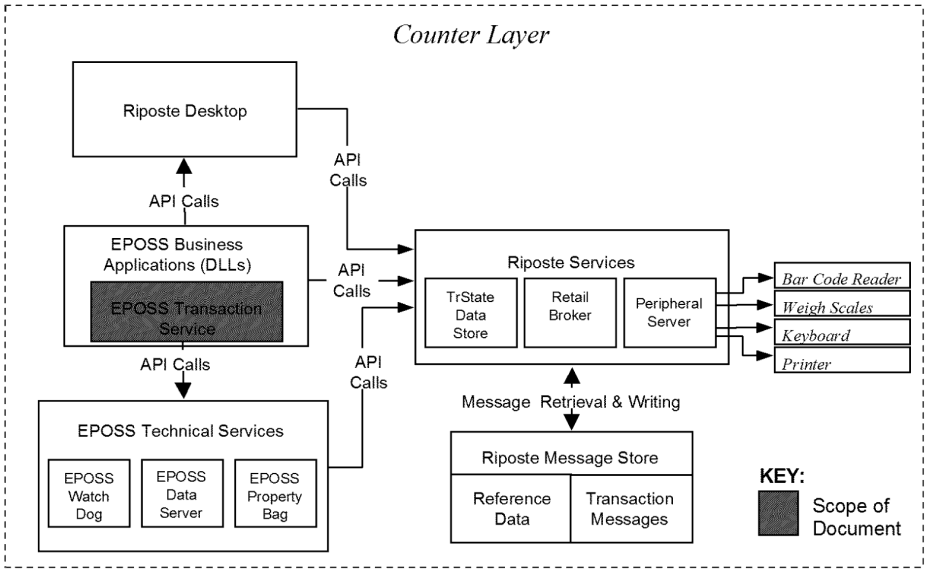


Figure 12- Associations between EPOSS Desktop Components

In terms of deployment, the EPOSS System consists of:

- One instance of the Riposte desktop
- Instances of Riposte and EPOSS Technical Services, e.g. Riposte Retail Broker, Riposte Peripheral Server, EPOSS Data Server, EPOSS Property Bag, EPOSS Watch Dog.
- Instances of EPOSS Applications
- Riposte Message Store

The desktop initialisation process requires the registration of the individual EPOSS Applications with the Desktop. This process ensures that the components are successfully created and in a defined sequence. This is necessary, as dependencies exist between the individual components. For example, the majority of EPOSS Applications, such as the EPOSS Core and EPOSS Settlement Object, use the functions provided by the Retail Broker through established API calls. Therefore, since a dependency (association) exists between two components and the initialisation procedure of the desktop verifies that the Retail Broker has been successfully created.

FUJ00171948

5.3 Component Design

5.3.1 EPOSS Context

A DFD is shown in Figure 13, which establishes the context within which the Transaction Service exists.

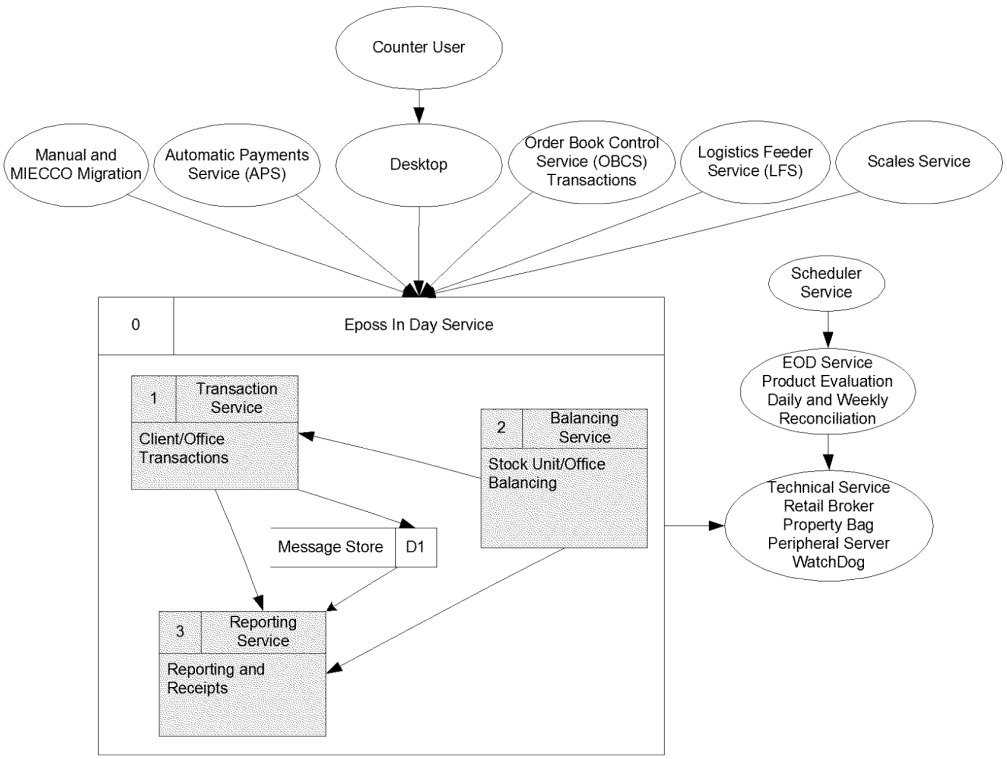


Figure 13 –EPOSS Context is part of In Day Services

FUJ00171948

5.3.2 Transaction Service

5.3.2.1 User Perspective

The user perspective of the Transaction Service is used to identify and name the generic components of the system. There are four representative elements identified by Presentation, User, Service and Transaction, illustrated in Figure 14.

Presentation denotes user interface entities managed or displayed via the Desktop Service. User represents the start and end of user sessions and any underlying alternate sessions initiated by the suspend control.

Service denotes the type of business session requested by the user also known internally as the Session Mode. A (Transaction) Session represents the set of business transactions that must balance before committing to the message store.

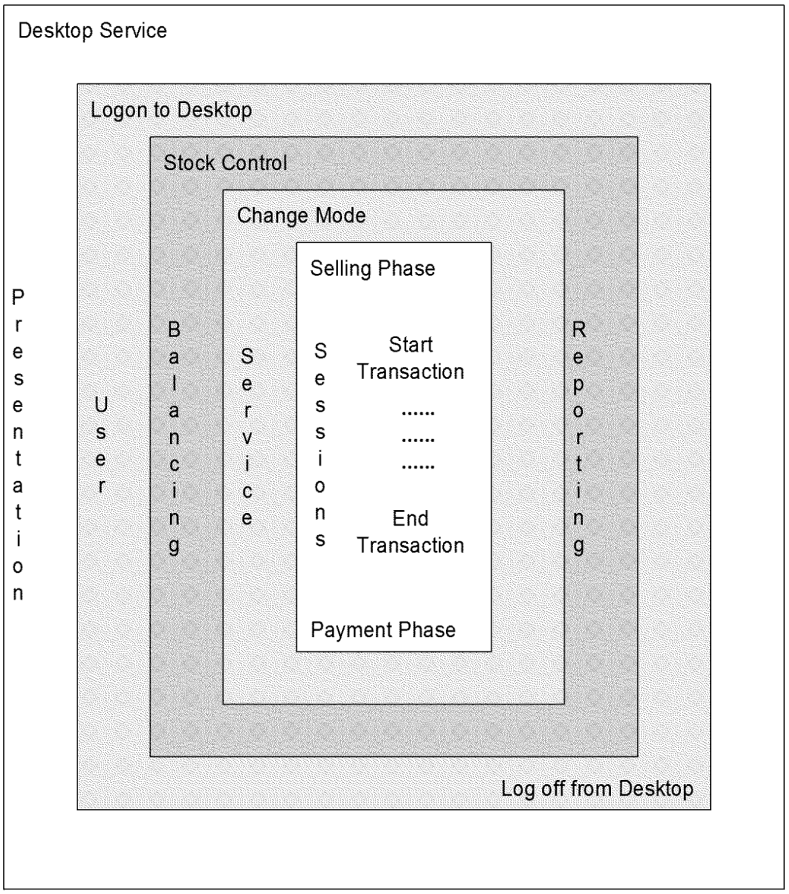


Figure 14 – User Perspective of Transaction Service

FUJ00171948

5.3.2.2 Development Perspective

The development perspective of the Transaction Service is used to establish a descriptive name for the generic components of the system. There are four representative elements identified by Desktop Presentation, User Session, Service Request and Transaction Session, illustrated in Figure 15.

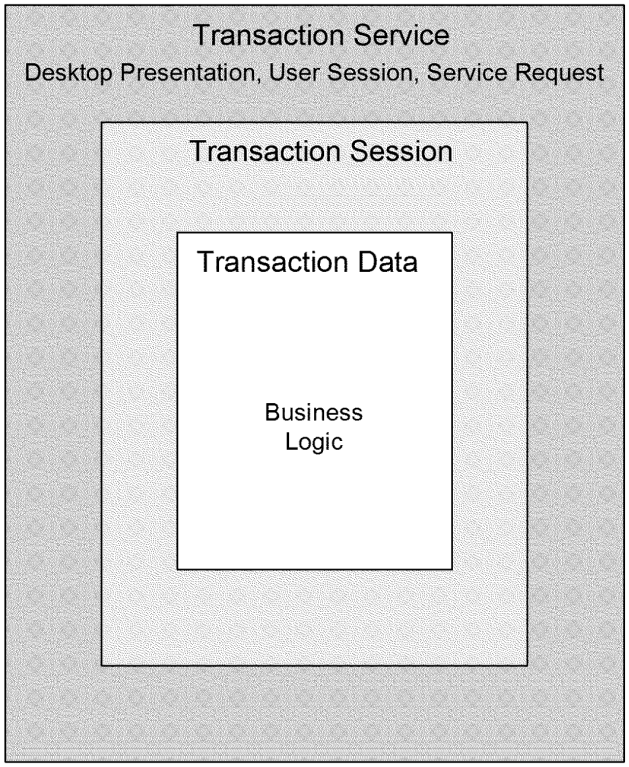


Figure 15 - Development Perspective of Transaction Service

FUJ00171948

5.3.2.3 Internal Viewpoint

The entities managed by Desktop Presentation, User Session, Service Request and Transaction Session are given in Figure 16. Desktop Presentation categorises the different types of objects that can be manipulated with the Desktop Service. User Session, Service Request and Transaction Session directly identify specific components.

Desktop Presentation manages and displays menu screens, panel screens and controls on menu screens or panel screens. User Session controls logon and logout requests and provides suspend and swap routines for saving and restoring the state of a transaction session whenever the user switches between sessions at the counter.

Service Request is responsible for changing the session mode as directed by the user. This prepares the menu of items for the mode of operation requested. Transaction Session generates all the business transactions for the session mode set up by Service Request.

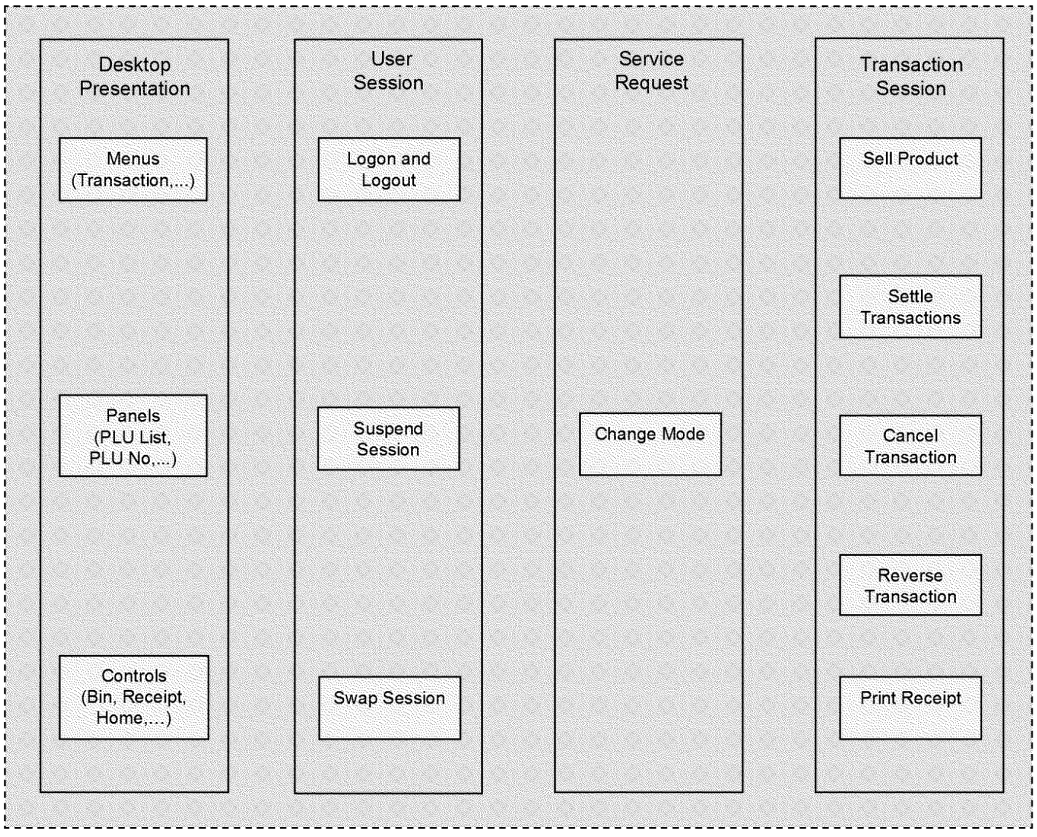


Figure 16 – Internal Viewpoint of Transaction Service

FUJ00171948

5.3.2.4 External Viewpoint

The external entities used by and external entities that utilise Transaction Service are illustrated inFigure 17. APS, OBCS, LFS, Scales Service and Balancing Service all need to utilise the Transaction Session component of Transaction Service. The user has direct access to the Transaction Session component via the Desktop Service. Transaction Service utilises the Reporting Service to control printing of receipts. The numbering system for components of Transaction Service is defined from this point onward, instead of starting from Figure 13, to minimise the number of levels introduced.

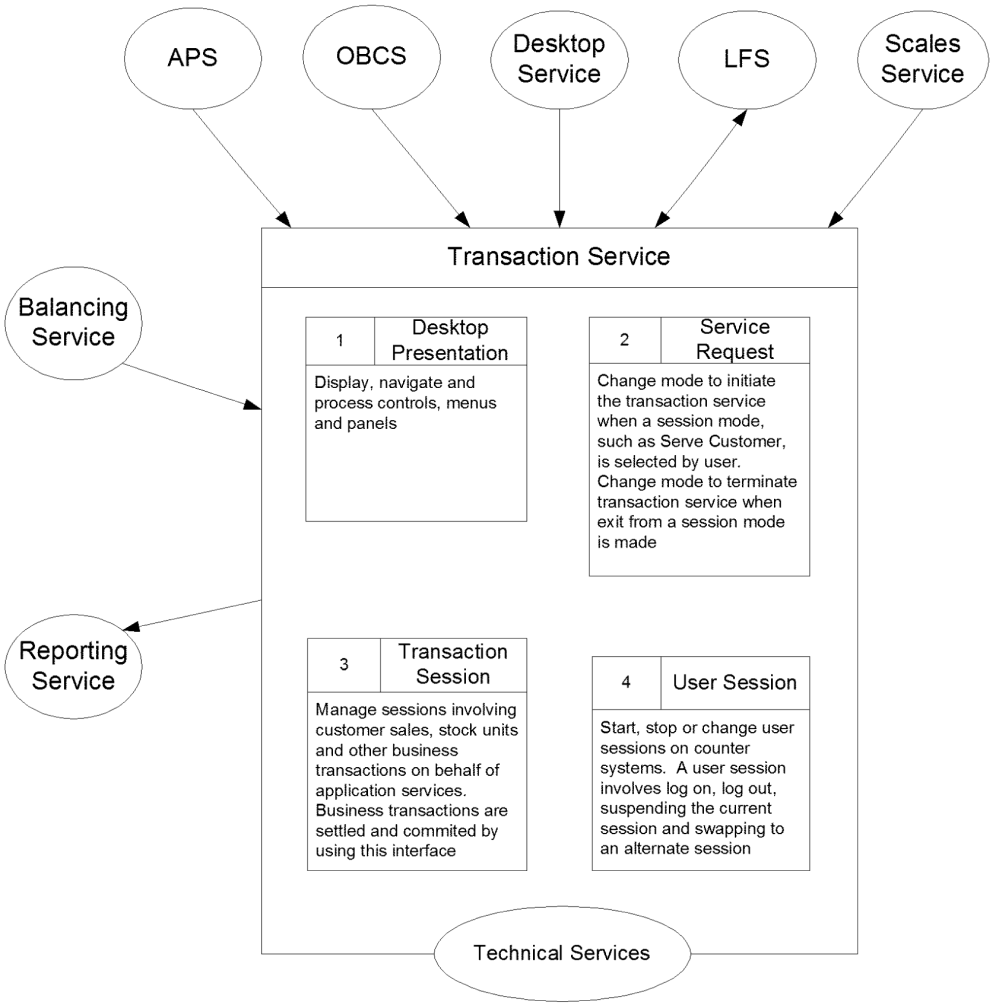


Figure 17 – External Viewpoint of Transaction Service

FUJ00171948

5.3.2.5 Application Components

An overview of the application components that comprise the Transaction Service is given in Figure 18. This introduces the numbering scheme for logical components and reflects the hierarchy of subsequent subsections of the document. Each subsection presents a logical abstract model of the underlying code.

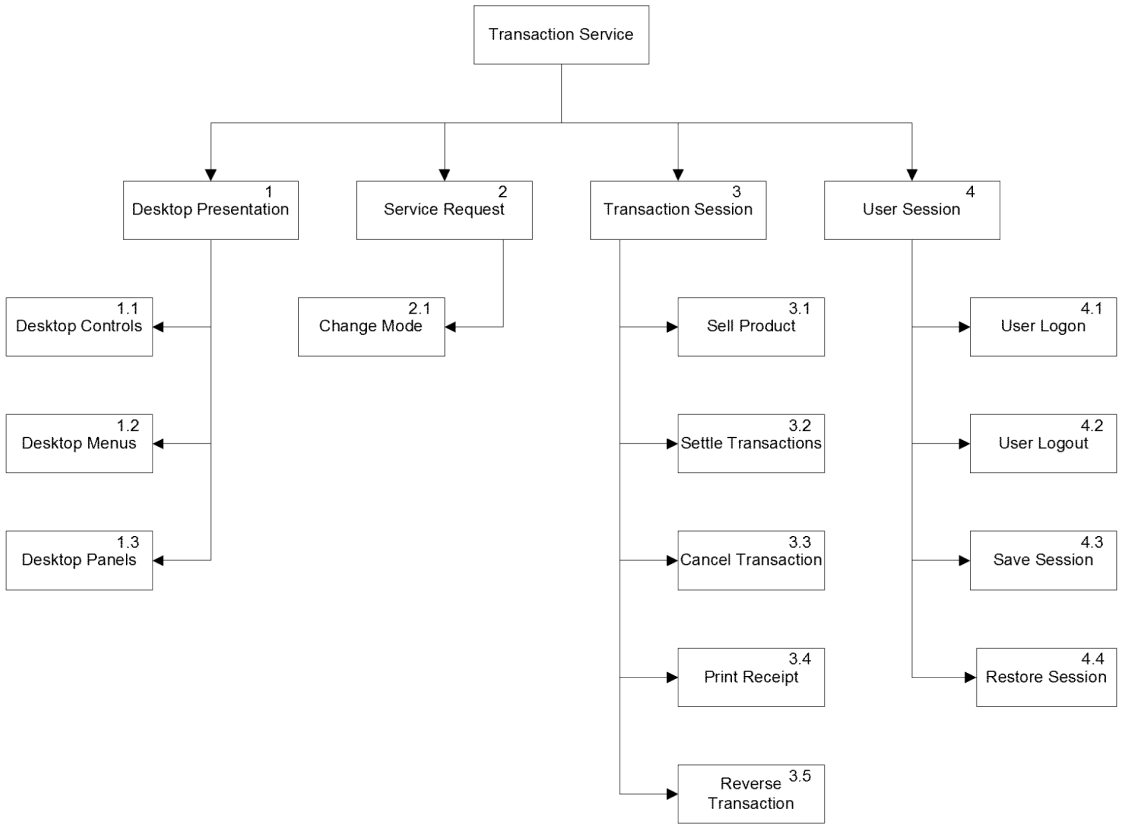


Figure 18 – Application Components of Transaction Service

FUJ00171948

5.3.3 Desktop Presentation

5.3.3.1 User Perspective

Figure 19 illustrate the specific entities managed by Desktop Presentation from the user perspective. Before the Transaction Service can be started the user must select a session mode to establish the type of business transactions required. For more details of the session mode see section 5.3.3.2, which explains the business perspective of Desktop Presentation.

The selection of a session mode displays a Product Menu containing a set of products or services for initiating a transaction session. A Method of Payments Menu is displayed when the user selects the Finish button during a Serve Customer, Reversal or Housekeeping session. Desktop Menus is responsible for displaying these menus.

Navigation controls are presented on menu screens and panel screens for returning to a previous menu. Function controls include the Bin and Quantity buttons on product menus and Qty Shop, Receipt Reprint and Receipt buttons on the Functions menu. Desktop Controls is responsible for processing the aforementioned controls.

From the Functions menu the user can select a PLU List to select a product from a set of products presented in a pick list. The user can also select PLU No, from the Functions menu. Panels are also displayed when the user selects the Quantity control, Qty Shop or any one of the product groups presented within the Local Schemes menu in a Serve Customer session. Account, Quantity and Price panels may also be appear when transacting an item during a transaction session. The user can also access the PLU No and PLU List panels by pressing a hotkey assigned to the menu buttons. Desktop Panels is responsible for displaying and processing these panels.

FUJ00171948

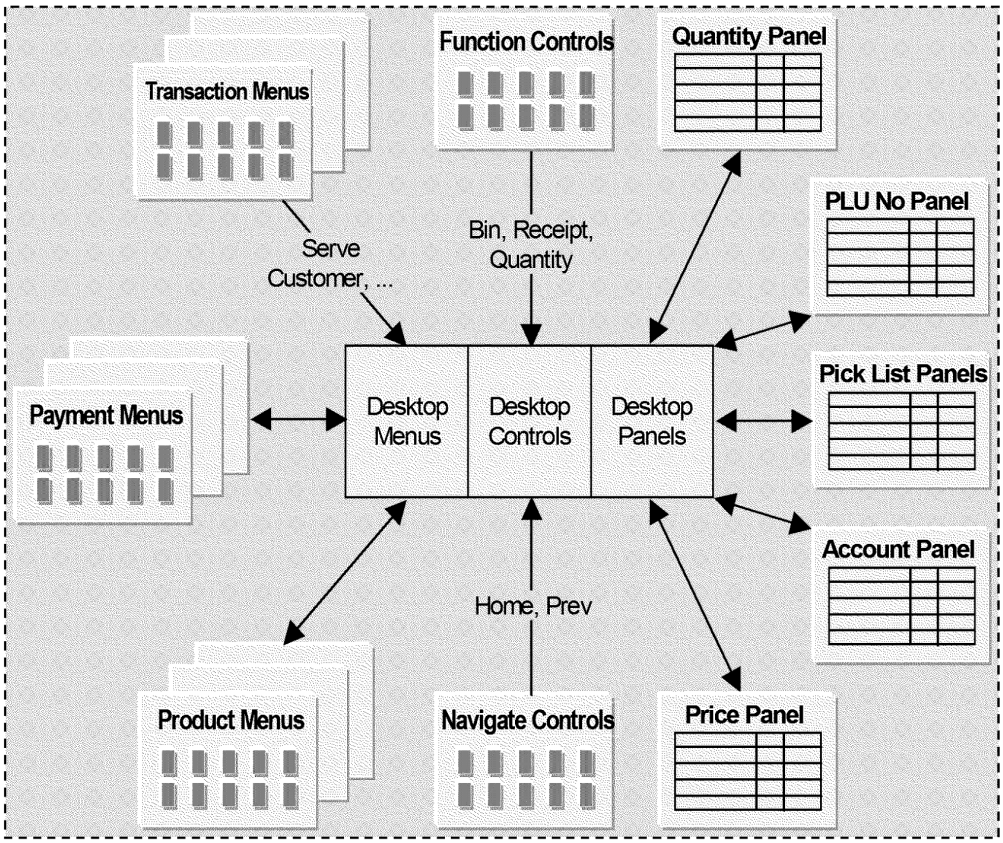


Figure 19 – User Perspective of Desktop Presentation

FUJ00171948

5.3.3.2 Business Perspective

The table in Figure 20 defines the types of transaction that are managed by Desktop Presentation from the business perspective. The submenu title and button caption for each session mode is listed. There are several types of mode associated with a transfer, remittance, revaluation and reversal.

The user initiates the Transaction Service when the type of business transaction, known as the session mode, is selected from the Transaction menu or one of its submenus. The Transaction menu is displayed when the user chooses the Transaction button on the home menu.

Each session mode is associated with a specific button on the Transaction menu or one of its submenus. Serve Customer, Housekeeping, Bulk Input, Non-Account Data and Parcel Traffic can be selected directly from the Transaction menu. Otherwise the Transfers, Remittance, Reversals and Revaluation buttons on the Transaction menu are used to navigate to any one of the other session modes.

All transactions at a counter position involving a customer session are managed in Serve Customer mode. APS, OBCS and Scales applications are transacted in this mode. Parcel Traffic mode is used later to send a batch of one or more parcels, transacted in serve Customer mode, to one or more locations.

A Transfer mode enables the transfer of stock items between stock units assigned to a Post Office outlet. A Remittance mode enables stock items, belonging to assigned stock units, to be transferred to and from an outside source such as another Post Office outlet. Stock items can also have their value changed for accounting purposes using the Revaluation mode.

Bulk Input (Recovery) is a session mode reserved for recovery from an outage at a Post Office outlet. When there is a loss of service, for any considerable period of time, all transactions at a Post Office Counter are recorded manually. When normal service resumes the clerk can input all transactions recorded manually into the system via Bulk Input.

A single transaction, settled during a transaction session, can be reversed, using the known transaction identifier, in Existing Reversal (ER) session mode. If the transaction identity is unknown, when a receipt is unavailable, the user can reverse a transaction within a New Reversal (RV) session mode, which transacts sale items with negated sale values. This is like a sales transaction but the values of each product transacted are negated so that a refund can be given.

A transfer session cannot be reversed in any of the above session modes. Instead, a transfer out session can only be reversed within an Existing Reversal (ER) session mode. Every transaction within the transfer out session is reversed. A transfer in session is irreversible other than via a transfer out session.

Housekeeping mode is used to settle the accounting system and correct errors made during transaction sessions. Discrepancies arising from remittance sessions, such as a shortage or surplus of stock received from another outlet, are recorded during housekeeping sessions.

Inventory items associated with special types of service, such as fishing licences, are handled in Non-Accounting Data mode. These types of transaction are used to determine stock levels and volumes of business for each type of service.

FUJ00171948

Session Mode	Button Caption	Menu Title
SC	Serve Customer	Transactions
TI	Transfer In	Transfers
TO	Transfer Out	Transfers
RIAD	Auto-Distribution	Remittance In
ROAD	Auto-Distribution	Remittance Out
RICL	Client	Remittance In
ROCL	Client	Remittance Out
RIOP	Other Post Office	Remittance In
ROOP	Other Post Office	Remittance Out
RISD	Supply Division	Remittance In
ROSD	Supply Division	Remittance Out
RODC	Data Processing Centre	Remittance Out
ER	Existing	Reversals
EV	Not In Session	Reversals
RV	In Session	Reversals
RU	Up	Revaluation Up
RD	Down	Revaluation Down
NAD	Non-Account Data	Transactions
HK	Housekeeping	Transactions
PT	Parcel Traffic	Transactions
REC	Bulk Input (Recovery)	Transaction

Figure 20 – Table of Session Modes and their associated Menu Buttons

FUJ00171948

5.3.3.3 Development Perspective

Figure 21 illustrates and describes the generic entities managed by Desktop Presentation from the development perspective. Desktop Controls are buttons on menu or panel screens that are used to perform some special action during the current transaction session. Desktop Menus and Desktop Panels represent the set of menus and panels managed by the Transaction Service. The diagram also shows that the underlying entities are independent of any external inputs or outputs apart from the Desktop Service.

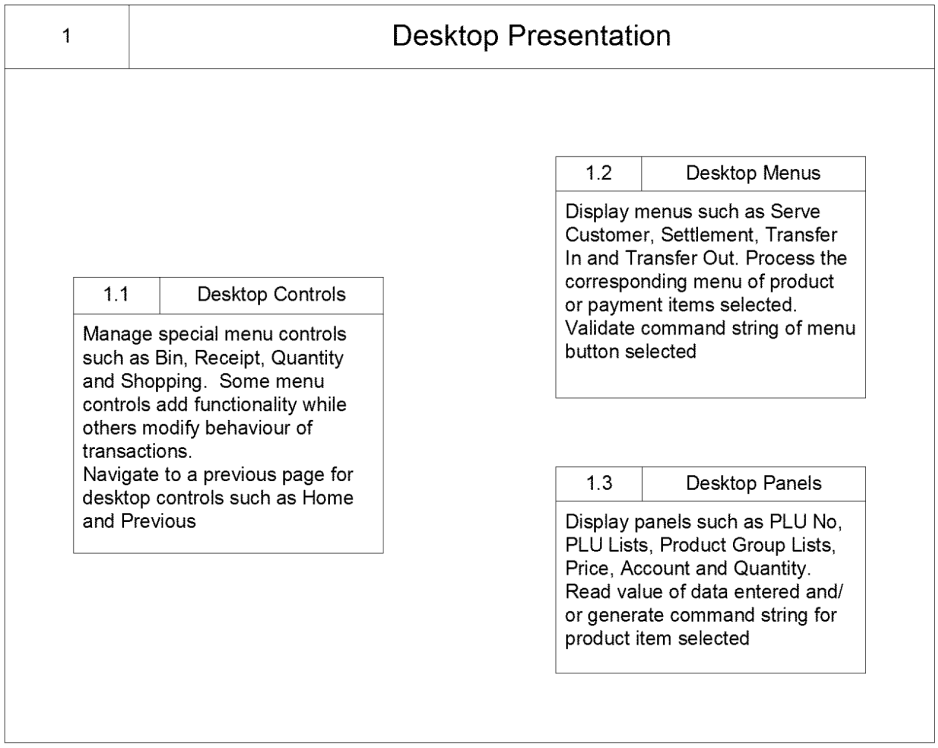


Figure 21 – Entities managed by Desktop Presentation

FUJ00171948

5.3.3.4 Screen Controls

Figure 22 illustrates and describes the specific entities managed by Desktop Controls. A Desktop Control is a button on a menu or panel screen used to perform some special action during the current transaction session

Navigation controls such as home and previous provide impulses from menu or panel screens whenever the Transaction Service needs to be informed. This is achieved by an application identifying itself to the Desktop Service.

Receipt Control is a menu button on the Functions menu for managing the printing of receipts. A receipt can only be printed on behalf of the current or last transaction session committed to message store.

Bin Control is a special menu button to enable the cancellation of transactions presented on the Product Stack. Bin Control is managed solely by the Desktop Service to prevent accidental deletion of a transaction. A transaction is cancelled when an item is selected from the Product Stack and the Bin Control is highlighted.

Quantity Control is a special menu button for controlling the quantity of sale or payment items transacted. Shopping Control is a menu button on the Functions menu for calculating items that can be transacted for a given amount of money.

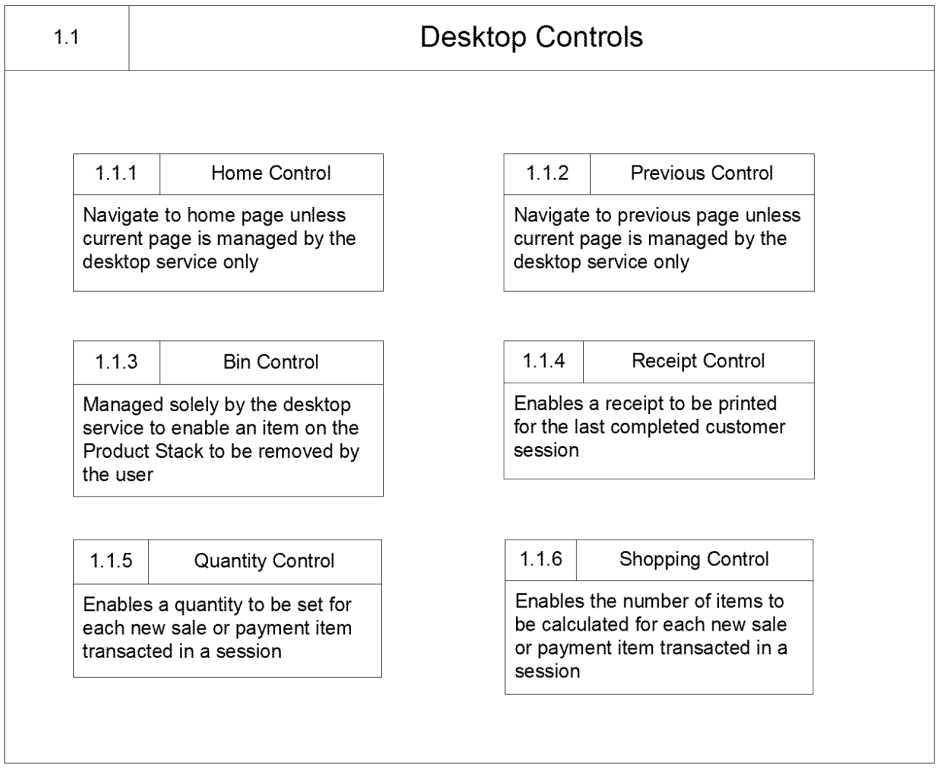


Figure 22 – Entities managed by Desktop Controls

FUJ00171948

5.3.3.5 Menu Screens

Figure 23 illustrates and describes the specific entities managed by Desktop Menus. The Transaction Service processes both sale and payment items on Transaction menus and navigates to the next menu when a new session mode is selected. A new session mode is initiated when one of the following menu items is selected:

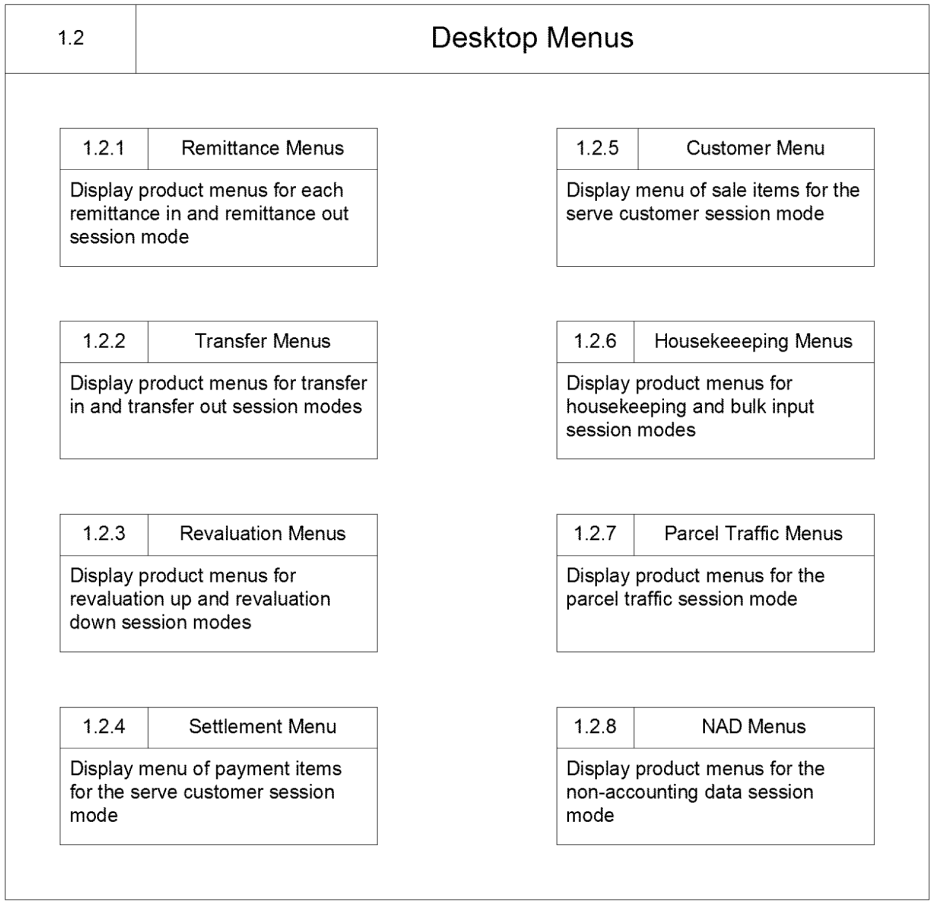


Figure 23 – Entities managed by Desktop Menus

FUJ00171948

5.3.3.6 Panel Screens

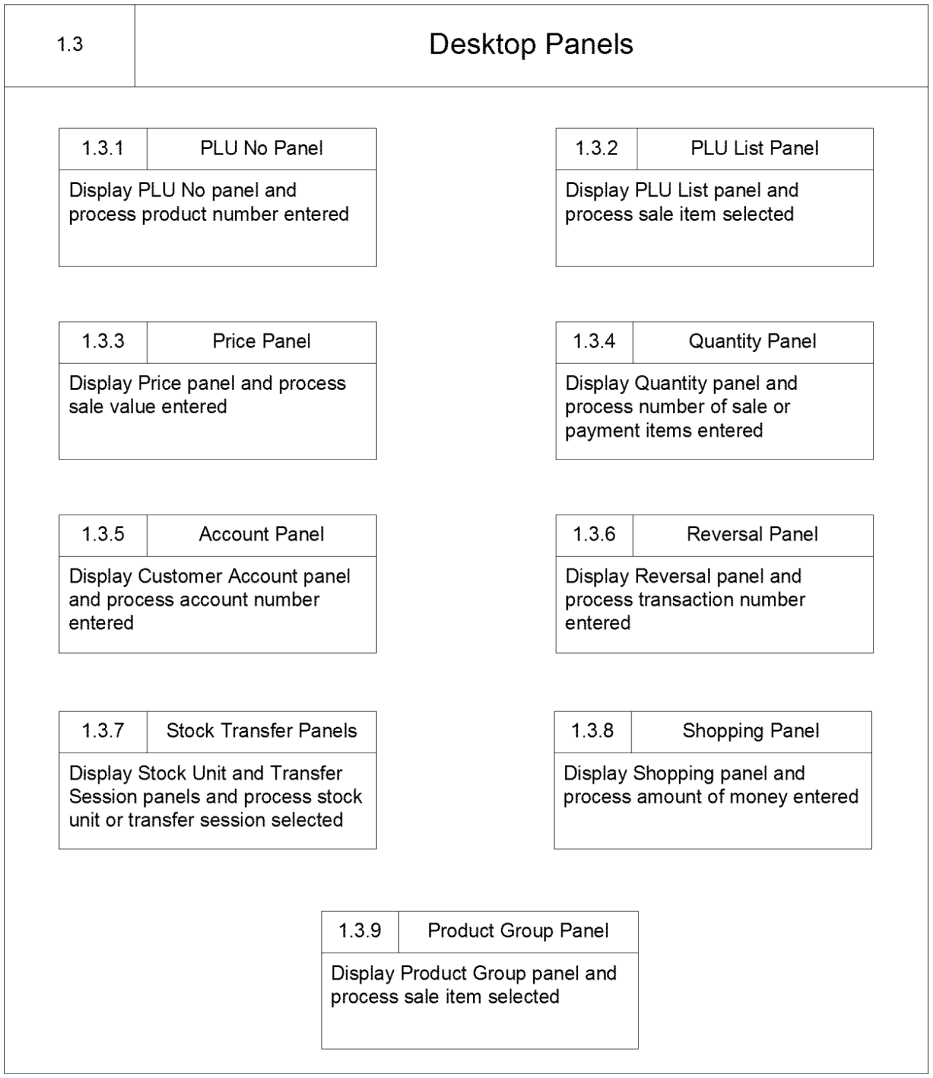


Figure 24 illustrates and describes the specific entities managed by Desktop Panels. There are only two types of panel, one that allows the user to select an item from a pick list and the other that presents the user with an edit box, with or without a numeric keypad, for entering a numeric value.

There are two types of pick list panel known as single selection and multiple selection. A single selection only allows one item at a time to be chosen from a given list of items. Multiple selection allows the user to select and change as many items as necessary to complete a transaction session. PLU List and Product Group panels, described below, are examples of single and multiple selection, respectively.

A Stock Unit panel enables the user to select a stock unit from a pick list during a transfer out session mode. A stock unit must be selected by the user to transfer out

FUJ00171948

items from an assigned stock unit to another stock unit allocated at the outlet. A user can accept items in a stock unit that have been transferred out during a transfer in operation, described below.

A Transfer Session panel enables the user to select a transfer out session from a pick list during a transfer in session mode. A transfer out session is selected by the user to transfer products into the user assigned stock unit. A user can preview each transfer out session before performing a transfer in operation.

The Qty Shop panel enables the user to switch between shopping and quantity modes for establishing the number of items involved in subsequent transactions during a customer session. The Shopping mode displays an alternative panel for establishing the number of items transacted in place of the Quantity panel.

PLU List and PLU No panels are used to transact a standard set of sale items during a customer or bulk input (recovery) session. A Product Group panel is also used to transact sale items, from the Local Schemes menu, during a customer session.

An Account panel is displayed when transacting an item that requires an account number such as a BT Gas bill. Quantity and Price panels are displayed when transacting an item that has an open-ended bulk quantity and sale value, respectively.

A Reversal panel enables the user to select a transaction number to reverse an existing transaction committed during a previous transaction session. The transaction number is determined from the receipt printed during the transaction session.

FUJ00171948

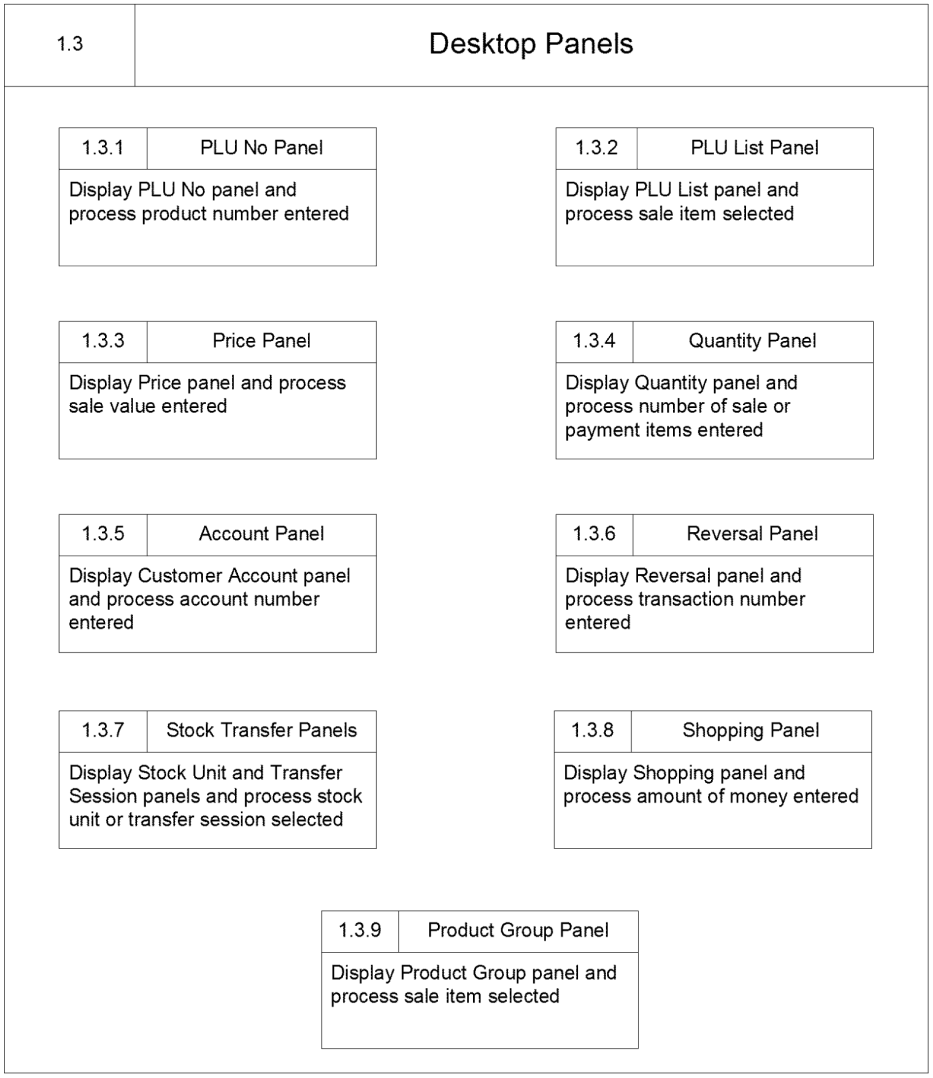


Figure 24 – Entities managed by Desktop Panels

FUJ00171948

5.3.3.7 Desktop Hierarchy

5.3.3.7.1 Transactions Menu

Transaction Service is initiated by one of the menus illustrated below. Some of the buttons on these menus establish the current session mode and the menu of products and services displayed. See Figure 25.

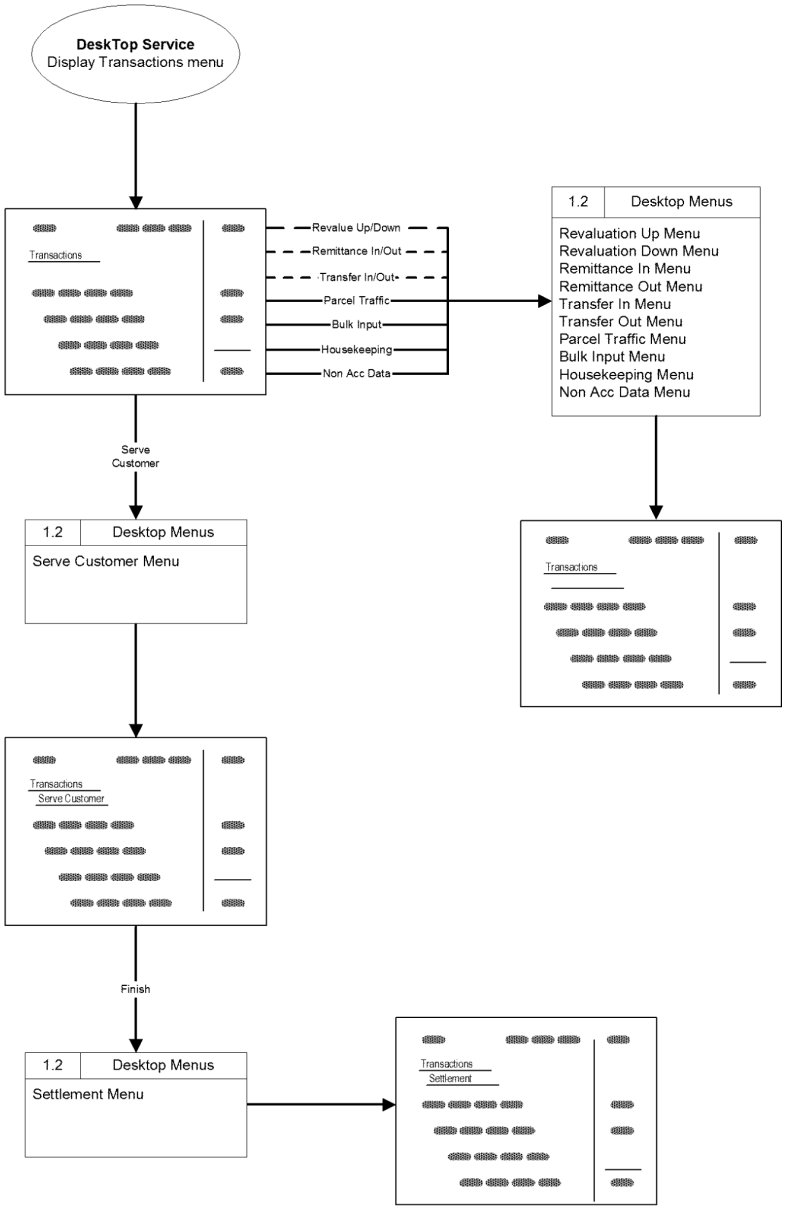


Figure 25 – Desktop Hierarchy from Transactions Menu

FUJ00171948

5.3.3.7.2 Functions Menu

Transaction Service manages several menu buttons on the Functions menu illustrated below. The captions on each menu button processed are entitled PLU No, PLU List, Qty Shop, Receipt and Reprint Receipt. See Figure 26.

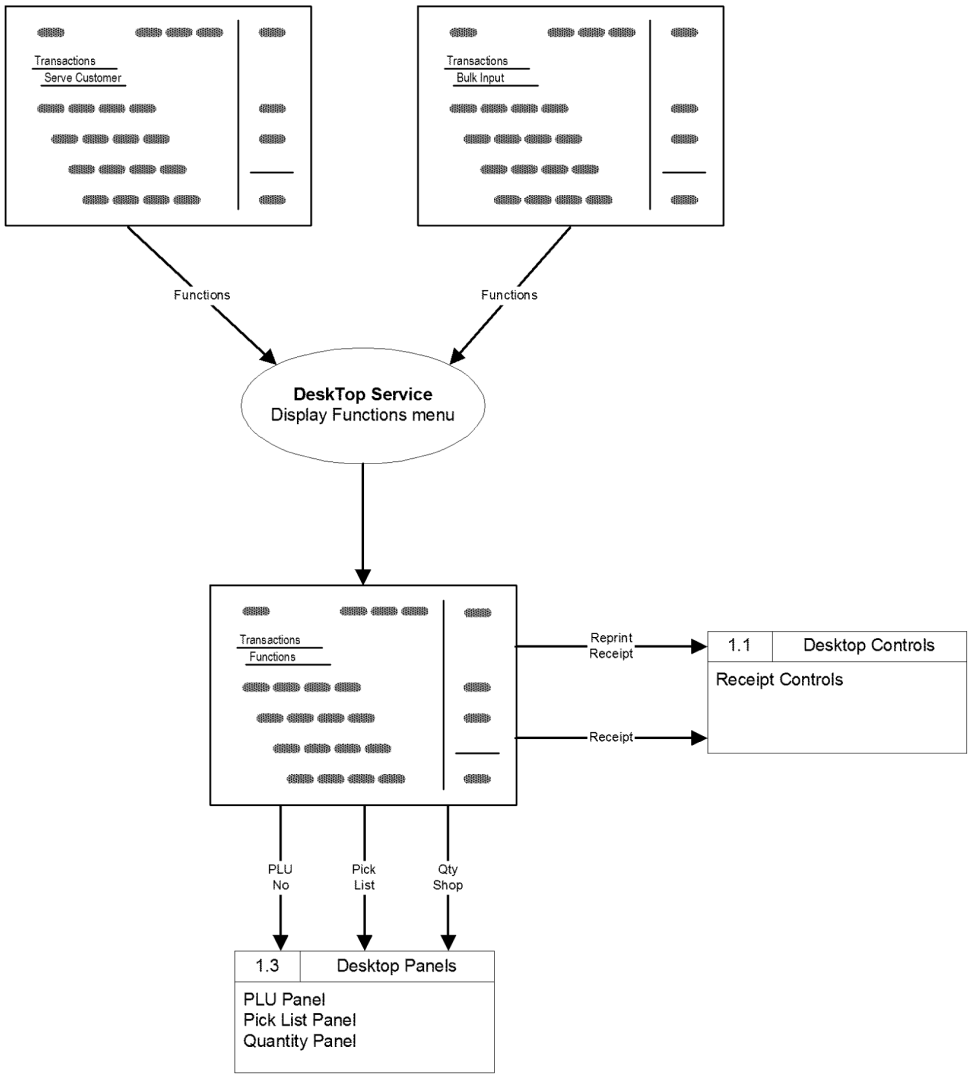


Figure 26 – Desktop Hierarchy from Functions menu

FUJ00171948

5.3.3.7.3 Reversals Menu

Transaction Service manages several menu buttons on the Reversals menu illustrated below. The captions on each menu button processed are entitled Existing, New and Transfer. An impulse is generated to reverse an identifiable customer transaction, an unknown customer transaction and a transfer out transaction, respectively. The remaining APS button is the responsibility of the APS application. See Figure 27.

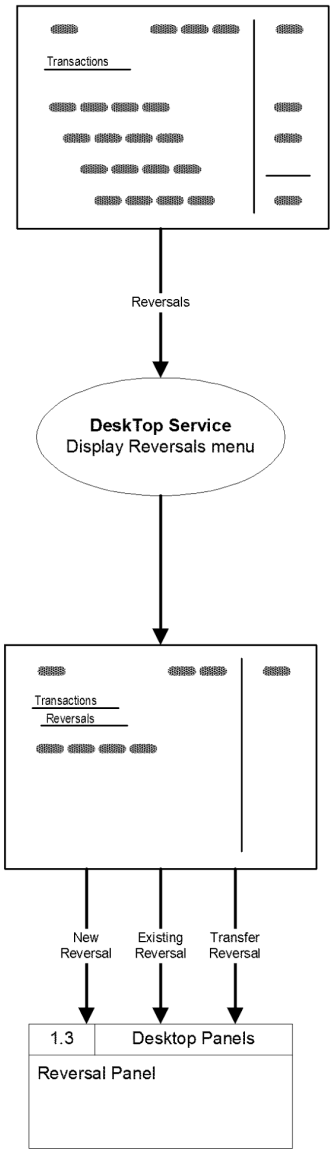


Figure 27 – Desktop Hierarchy from Reversals menu

FUJ00171948

5.3.3.8 Desktop Panels

5.3.3.8.1 PLU No Panel

5.3.3.8.1.1 External to Internal Mapping

The PLU No button on the Functions menu generates an impulse for the Transaction Service to display and process the PLU No panel. See Figure 28.

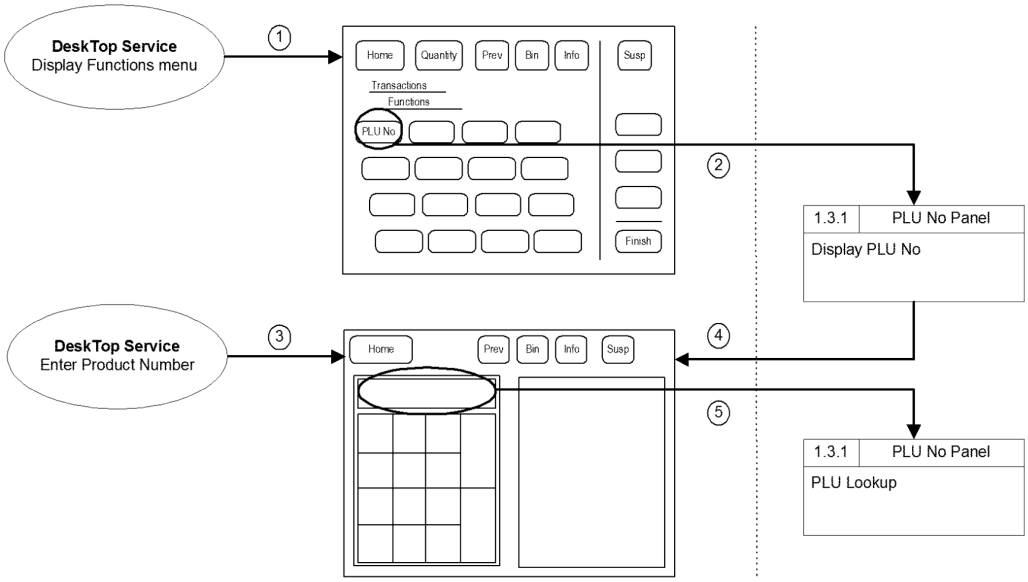


Figure 28 – External to Internal Mapping of PLU No Panel

FUJ00171948

5.3.3.8.1.2 *Computational Model*

The computational model for displaying the PLU No panel and processing the value entered by the user is given in Figure 29. A PLU List generated from PLU Impulses in Reference Data during initialisation of the Transaction Service is used to check the validity of the product number entered.

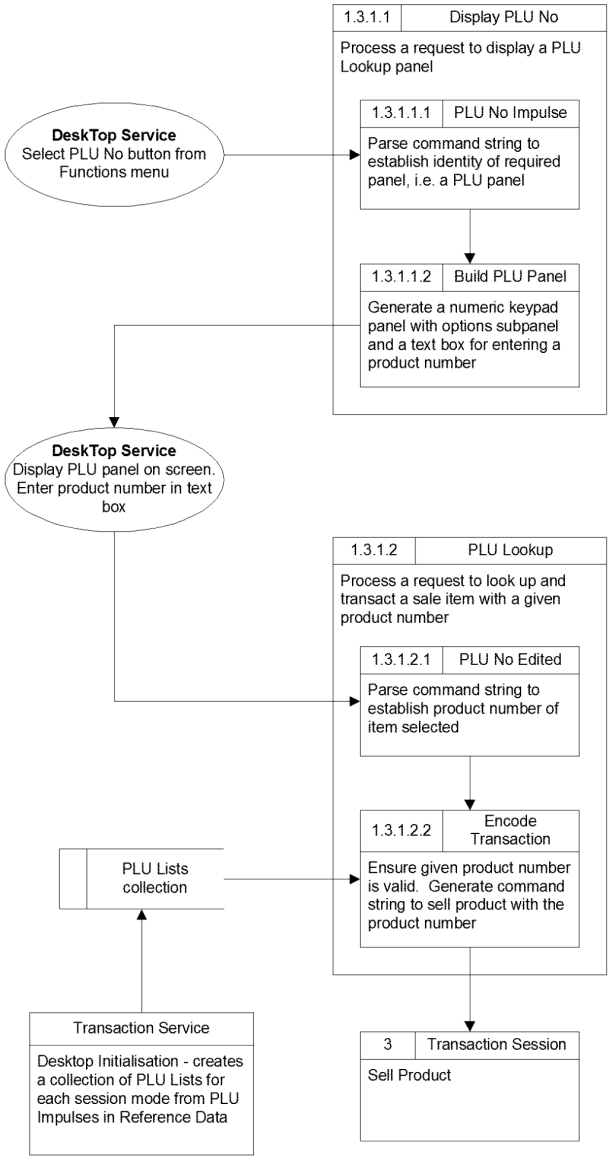


Figure 29 – Computational Model for PLU No Panel

FUJ00171948

5.3.3.8.2 PLU List Panel

5.3.3.8.2.1 External to Internal Mapping

The PLU List button on the Functions menu generates an impulse for the Transaction Service to display and process the PLU List panel. See Figure 30.

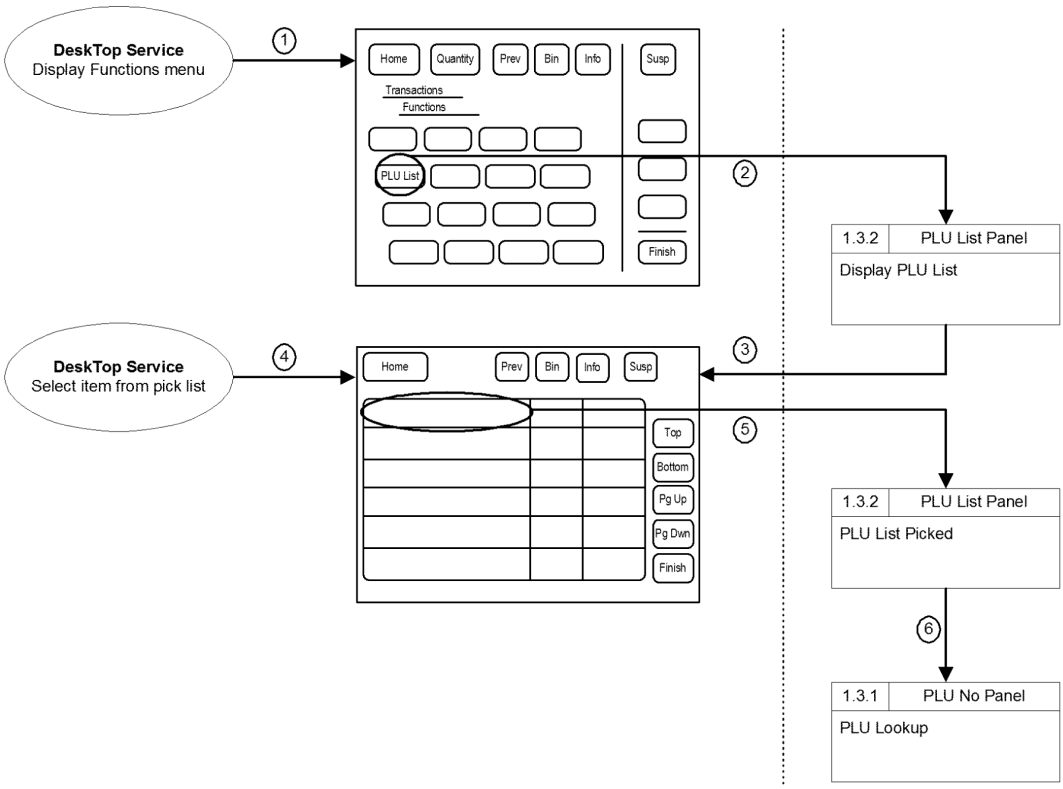


Figure 30 – External to Internal Mapping of PLU List Panel

FUJ00171948

5.3.3.8.2.2 Computational Model

The computational model for displaying the PLU List panel and processing the item selected by the user is given in Figure 31. A PLU List is generated from PLU Impulses in Reference Data during initialisation of the Transaction Service and is used to populate the pick list of products on the PLU List panel.

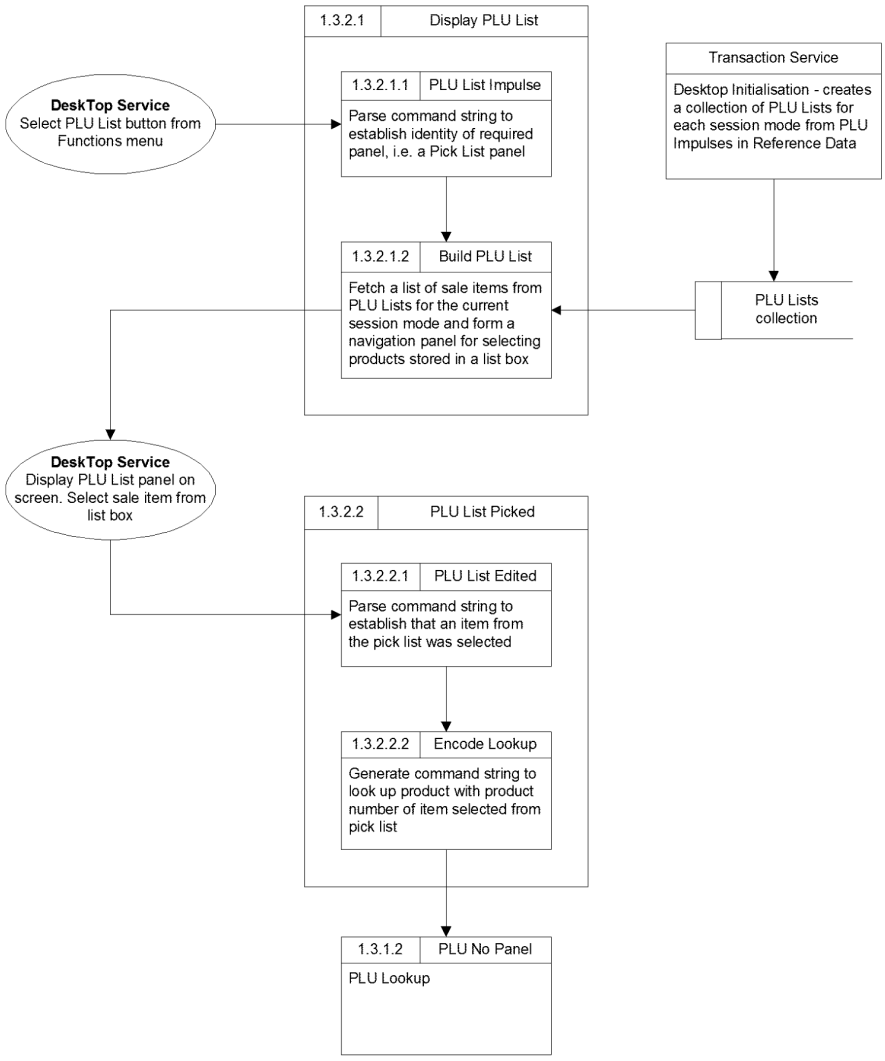


Figure 31 – Computational Model for PLU List Panel

FUJ00171948

5.3.3.8.3 Price Panel

5.3.3.8.3.1 Computational Model

The computational model for displaying the Price panel and processing the value entered by the user is given in Figure 32. A Price panel is displayed during the transaction of a sale item, with an open retail value, such as a BT Bill. All products and services defined by EPOSS Products, in Reference Data, without a fixed price will enforce this process when transacted.

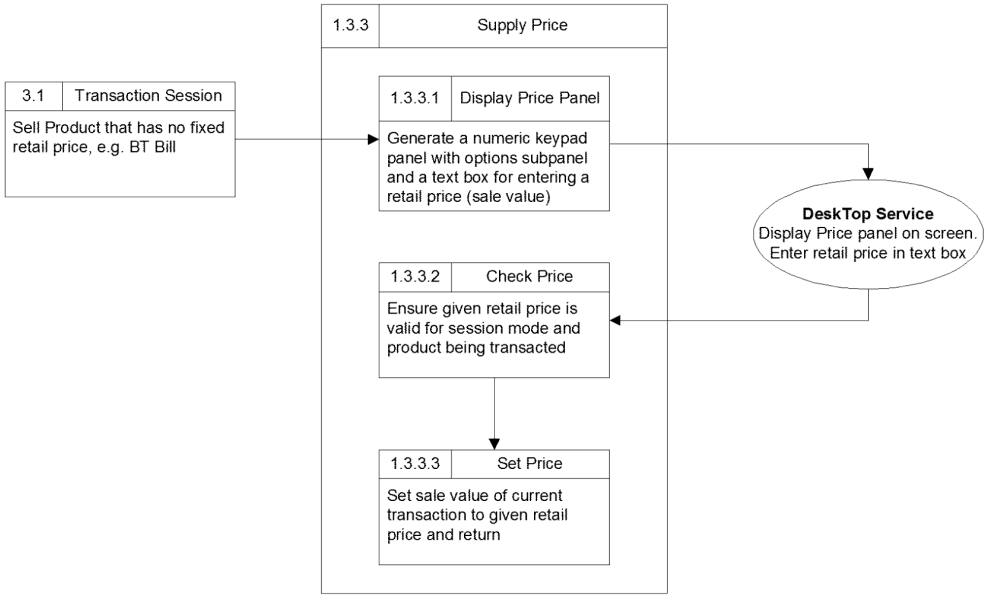


Figure 32 – Computational Model for Price Panel

FUJ00171948

5.3.3.8.4 Quantity Panel

5.3.3.8.4.1 Computational Model

The computational model for displaying the Quantity panel and processing the value entered by the user is given in Figure 33. A Quantity panel is displayed by selecting the Quantity control or during the transaction of a sale item without a fixed number of units. All products and services defined by EPOSS Products, in Reference Data, without a known quantity will enforce this process when transacted. The current value on the Quantity control is updated and extracted for subsequent transactions.

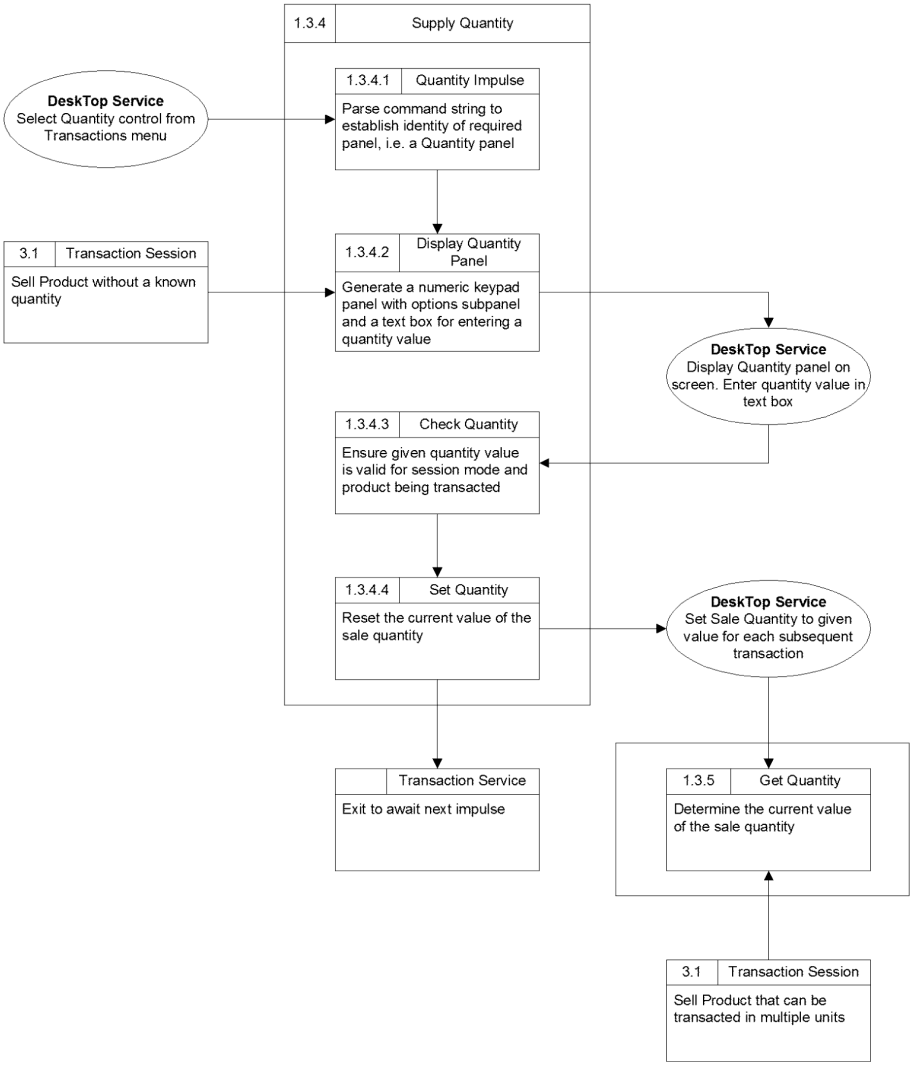


Figure 33 – Computational Model for Quantity Panel

FUJ00171948

5.3.3.8.5 Account Panel

5.3.3.8.5.1 Computational Model

The computational model for displaying the Account panel and processing the value entered by the user is given in Figure 34. An Account panel is displayed during the transaction of a sale item, with an account number, such as a BT Bill. All products and services defined by EPOSS Products, in Reference Data, with additional data fields will enforce a similar process when transacted. Additional data enables the execution of a script, which displays a panel for the user to provide extra information regarding the item being transacted.

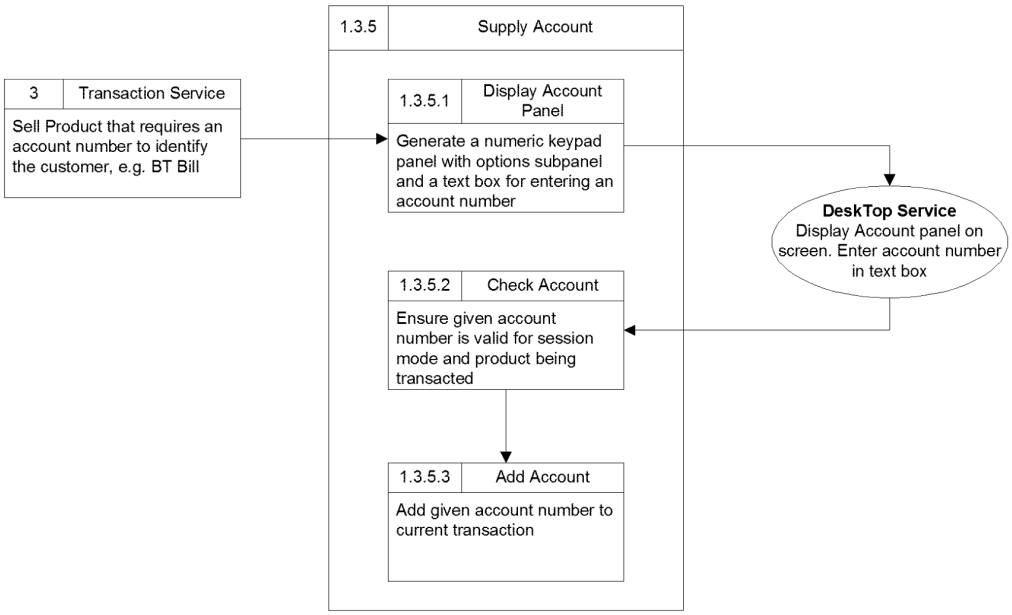


Figure 34 – Computational Model for Account Panel

FUJ00171948

5.3.3.8.6 Reversal Panel

5.3.3.8.6.1 External to Internal Mapping

The Existing button on the Reversals menu generates an impulse for the Transaction Service to display and process a Reversal panel. A Reversal panel enables the user to select a transaction number to reverse an existing transaction committed during a previous transaction session.

The Reversal panel consists of a numeric keypad panel with an options sub-panel and a text box for entering a numeric value. The transaction number is used to query the message store and find the committed transaction. If the existing transaction can be reversed a new transaction is generated to compensate and effectively annul the original transaction. See Figure 35.

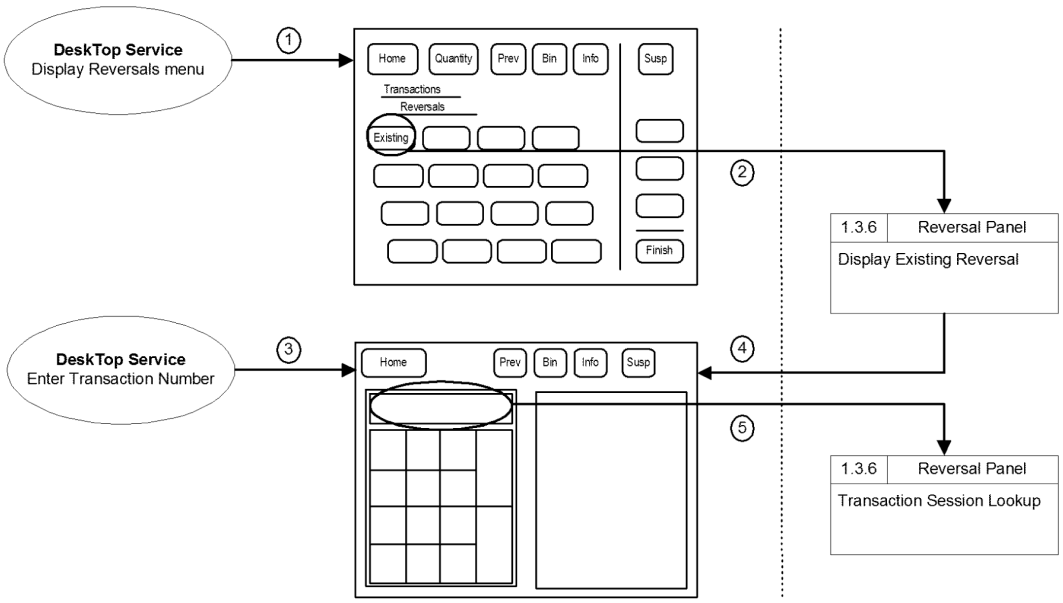


Figure 35 - External to Internal Mapping of Reversal Panel

The Transfer button on the Reversals menu also generates an impulse for the Transaction Service to display and process a Reversal panel. This panel is specifically designed for reversing transactions committed by a transfer out session. The New button on the Reversal menu is equivalent to starting a new transaction session for the current session mode.

FUJ00171948

5.3.3.8.7 Stock Transfer Panels

Management of Stock Transfer panels is an integral part of Service Request, which is described in section 5.3.4. When the transfer out session mode is selected, a Stock Unit panel is displayed to establish the target stock unit for transferring stock items from the user assigned stock unit. The selection of the transfer in session mode displays a Transfer Session panel for the user to select a transfer out session for transferring corresponding stock items into the user assigned stock unit.

FUJ00171948

5.3.3.8.8 Shopping Panel

5.3.3.8.8.1 External to Internal Mapping

The Qty Shop button on the Functions menu enables the user to select the shopping mode during a customer session. In shopping mode, a Shopping panel is displayed during the transaction of a product or service selected by the user. The user can switch between normal and shopping mode using the Quantity and Qty Shop controls, respectively.

The Shopping panel consists of a numeric keypad panel with an options sub-panel and a text box for entering an amount of money (total sale value). A quantity is determined for the current item selected from the given amount of money entered in the Shopping panel. The retail price of a single item is divided by the total sale value and truncated to give a whole number of items.

A panel of options is displayed giving the number of items calculated and the amount of change left over. The user is given an option to accept or adjust the number of items for an extra sum of money. When the quantity is accepted or adjusted by the user the required number of items is transacted via Transaction Session. The icon placed on the Product Stack will show the resultant number of items transacted and its total sale value. See Figure 36.

Completion of a transaction or the selection of Quantity control terminates the shopping mode. In this case, the ‘Shopping’ caption is cleared on the Quantity control to indicate that shopping mode is no longer available.

FUJ00171948

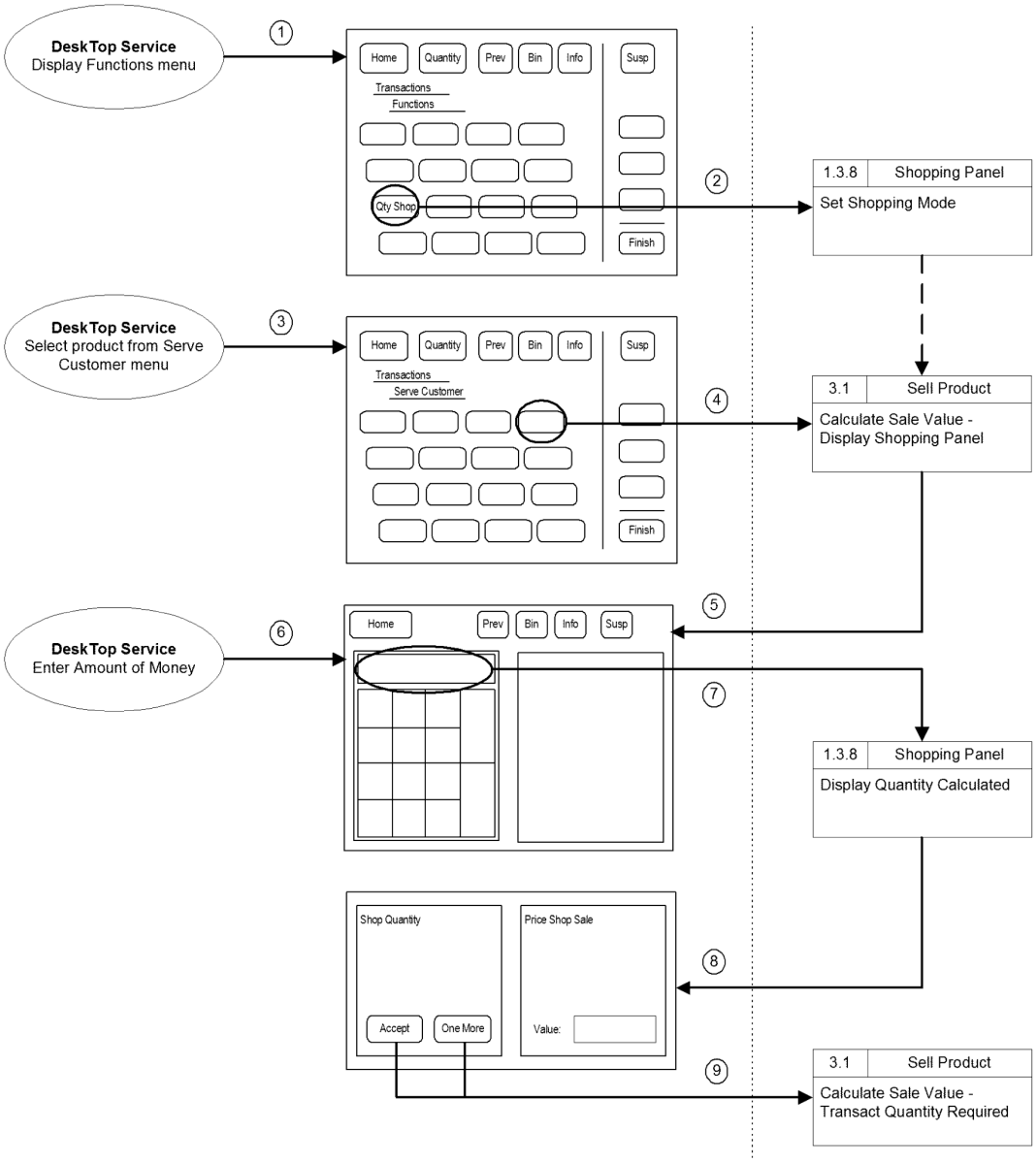


Figure 36 - External to Internal Mapping of Shopping Panel

FUJ00171948

5.3.3.8.9 Product Group Panels

5.3.3.8.9.1 External to Internal Mapping

The product group buttons on the Local Schemes menu, accessed from the Serve Customer menu, generate an impulse for the Transaction Service to display and process a product group panel. A product group panel is displayed when any one of the buttons on the Local Schemes menu is selected. This currently covers Travel Schemes, Home Care, Meals on Wheels, Rent Vouchers, Rent Cards, Council Tax Vouchers, Council Tax Cards, Election Schemes and Miscellaneous Schemes. See Figure 37.

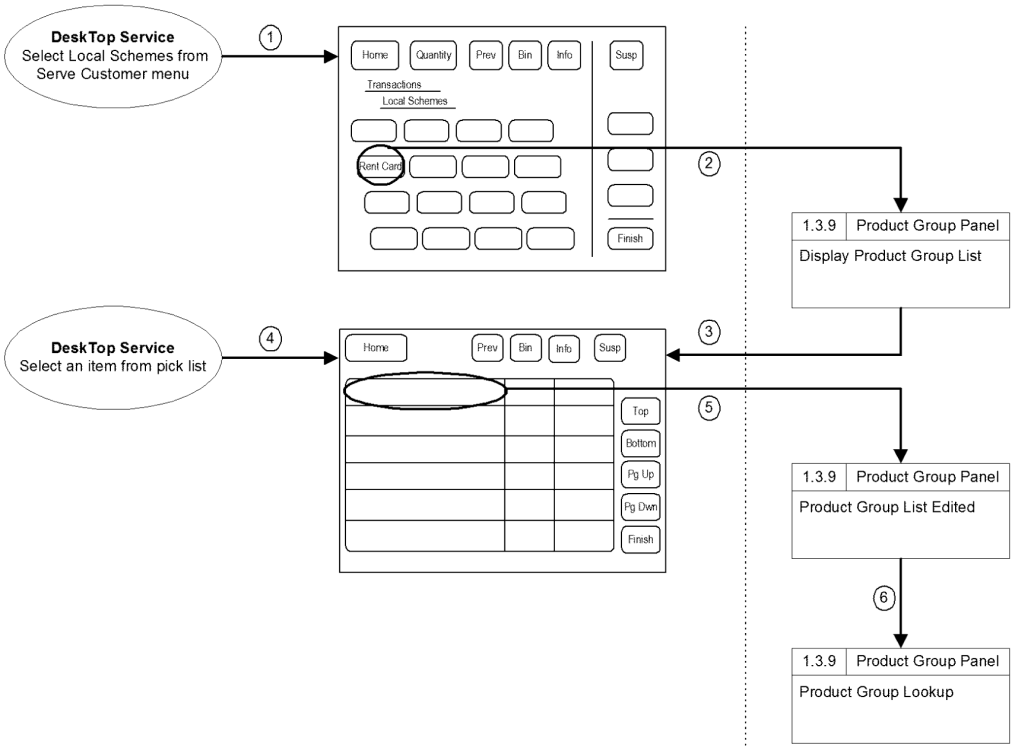


Figure 37 - External to Internal Mapping of Product Group Panel

FUJ00171948

5.3.3.8.9.2 Computational Model

The computational model for displaying a product group panel and processing the item selected by the user is given in Figure 38. A product group panel is generated directly from the Product Groups collection in Reference Data. A local collection of items is created and used to populate the pick list of products on the product group panel.

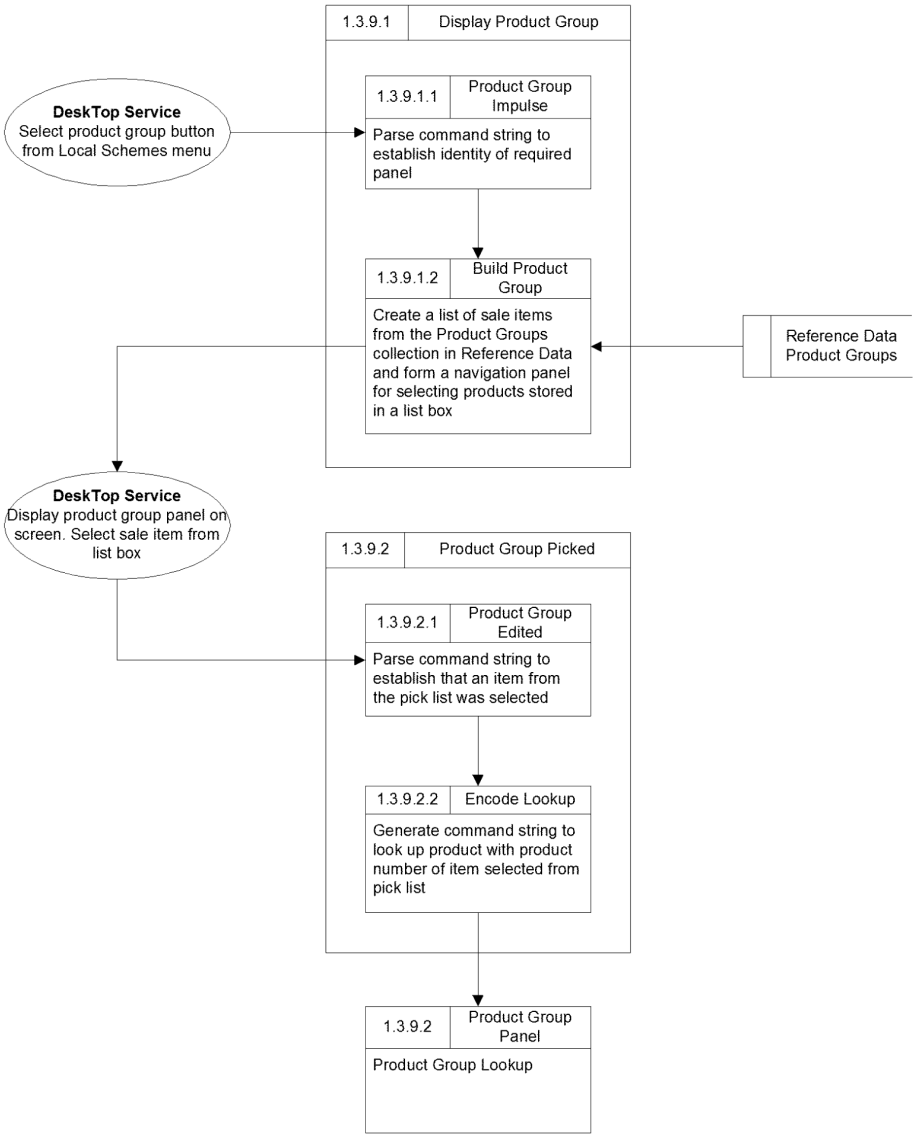


Figure 38– Computational Model for Product Group panels

FUJ00171948

5.3.3.9 Desktop Controls

5.3.3.9.1 Receipt Control

5.3.3.9.1.1 External to Internal Mapping

The Receipt and Reprint Receipt buttons on the Functions menu generate an impulse for the Transaction Service to print a new or print a duplicate receipt for the current transaction session. A receipt can only be printed when the current transaction session has been settled and committed to message store. See Figure 39.

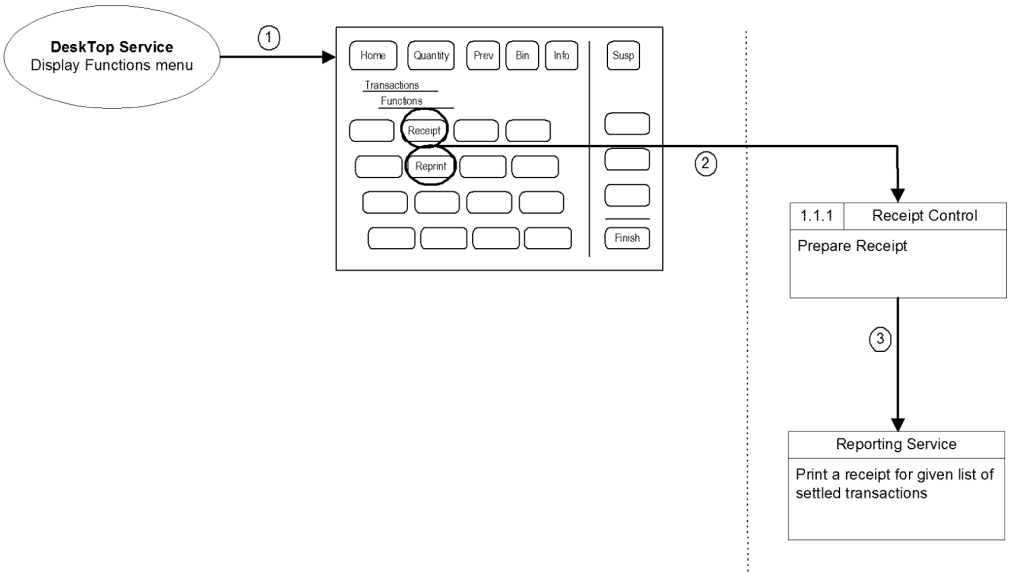


Figure 39 – External to Internal Mapping of Receipt Control

FUJ00171948

5.3.3.9.1.2 Computational Model

The computational model for printing a receipt for the current transaction session committed to the message store is given in Figure 40. A request to print or reprint a receipt is sent to the Reporting Service with a collection of the products and services transacted.

During the settlement of a transaction session a compulsory receipt may be required. In this case a similar list is created and a request to print a receipt is also sent to the Reporting Service.

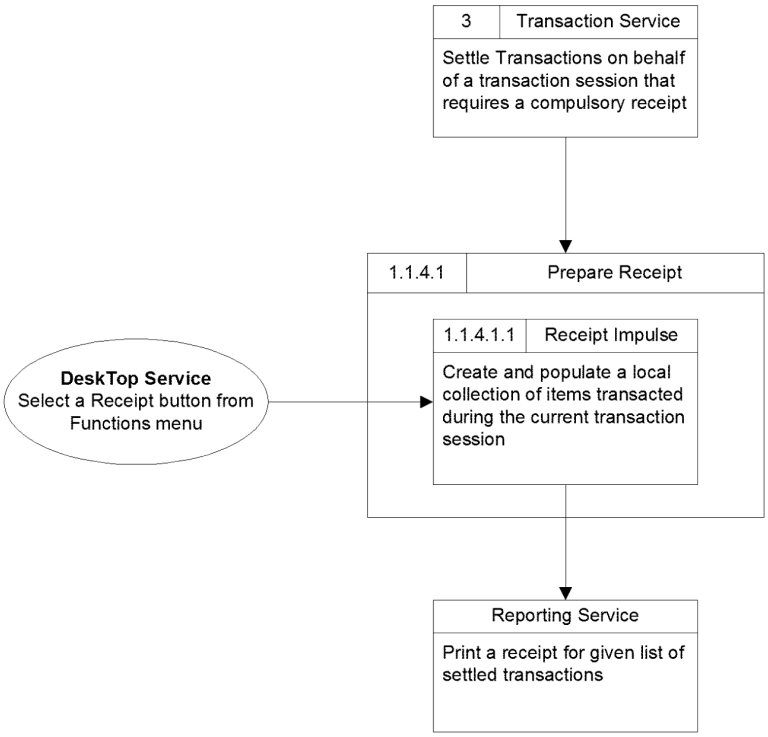


Figure 40 – Computational Model for Receipt Control

FUJ00171948

5.3.3.9.2 Bin Control

5.3.3.9.2.1 External to Internal Mapping

The Bin control on a Transactions menu enables the user to remove a transaction on the Product Stack during a transaction session. After highlighting the Bin control the user can choose one of the transactions on the Product Stack for deletion. When a transaction is successfully cancelled the session balance on the Finish button is automatically recalculated by the Desktop Service. The sale value of the sale or payment item removed is subtracted from the session balance. See Figure 41.

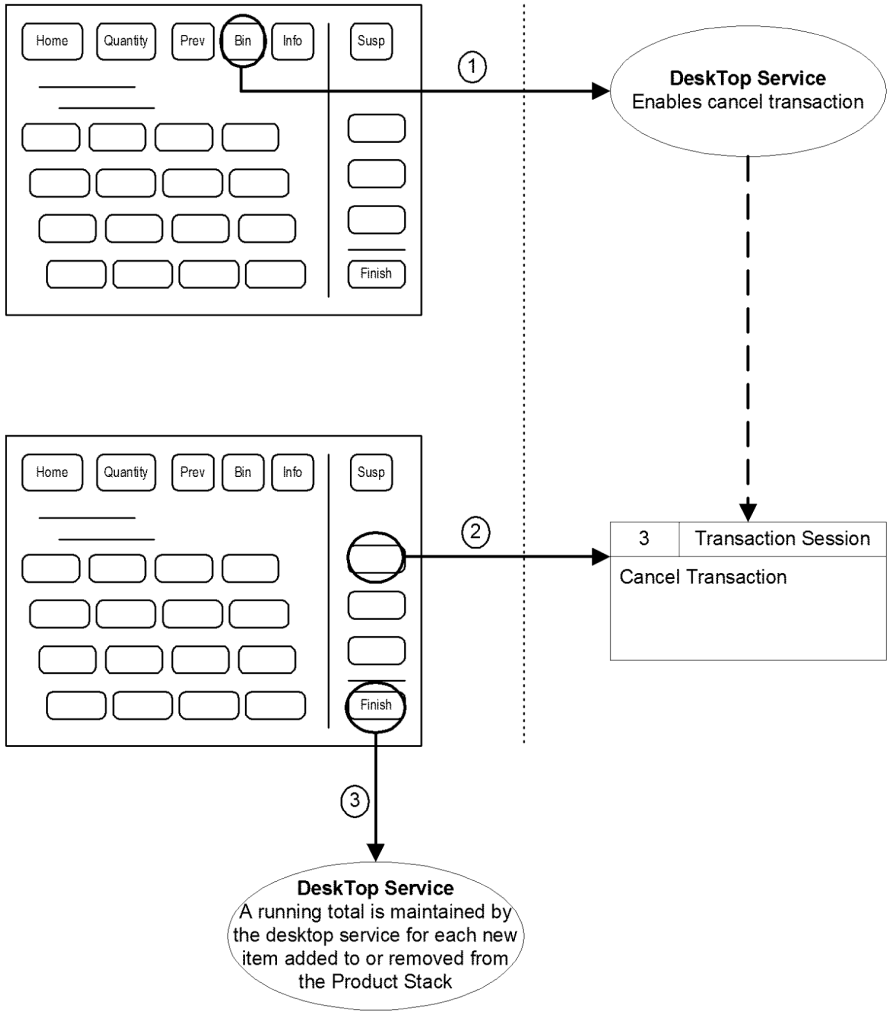


Figure 41 – External to Internal Mapping of Bin Control

FUJ00171948

5.3.3.9.2.2 *Computational Model*

The computational model for removing a transaction from the Product Stack during a transaction session is given in Figure 42. The details of the item and linked items are extracted from EPOSS Products. Cancellation of the item is rejected unless the item and all linked items can be voided. All linked items are removed before the selected item.

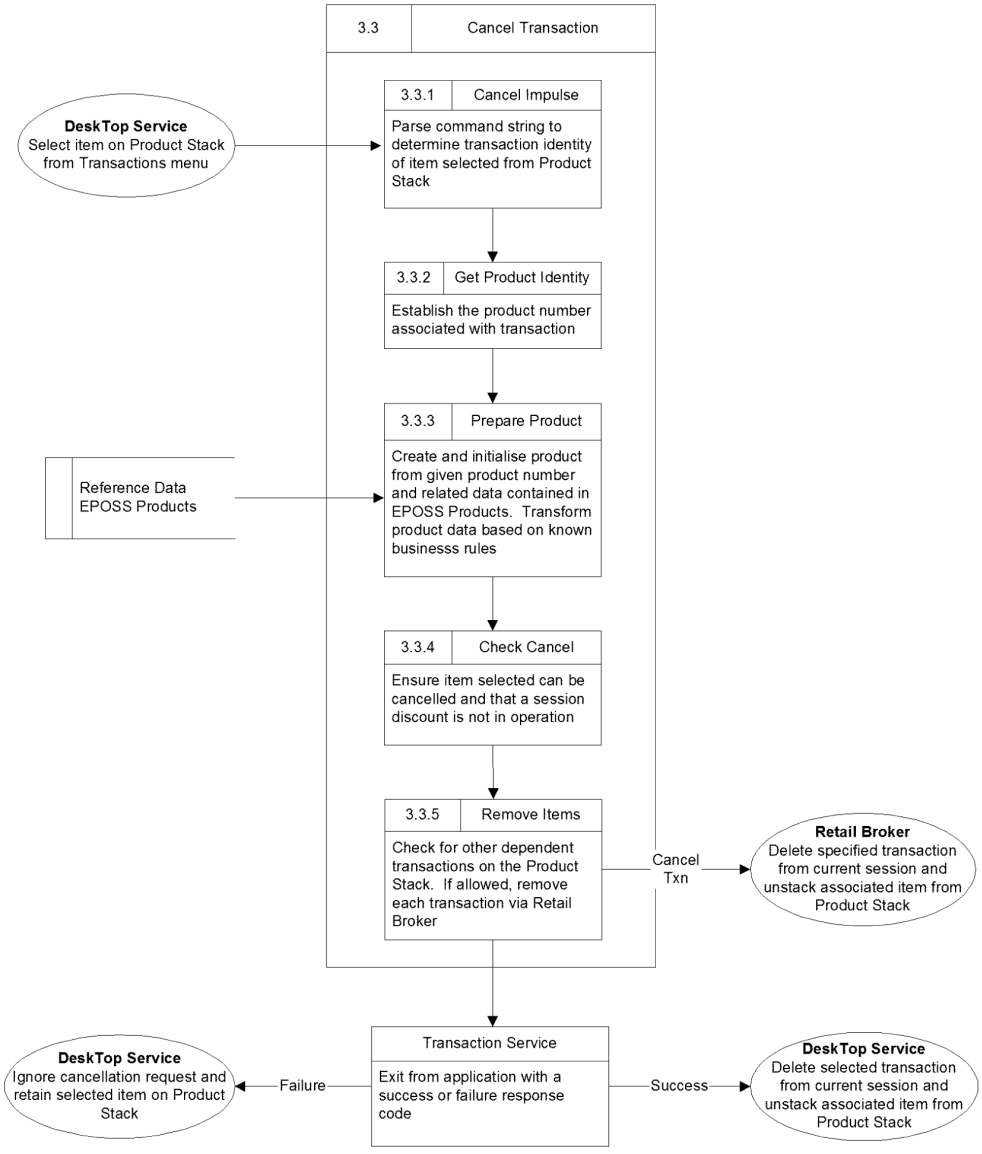


Figure 42 – Computational Model for Bin Control

FUJ00171948

5.3.3.9.3 Navigation Controls

5.3.3.9.3.1 External to Internal Mapping

5.3.3.9.3.1.1 Initiating Navigation Control

An application identifies itself, during normal operation, using the DLL name registered with the Desktop Service during the initialisation of the system. Every time an application runs the opportunity is available to take or relinquish control of the navigation system by setting or clearing the application name held in the desktop service.

Transaction Service always identifies itself when called but control is normally passed back to the Desktop Service when a menu or panel is displayed. After setting the application name in the Desktop Service the Transaction Service receives an impulse to process a navigation control. This process is illustrated in Figure 43.

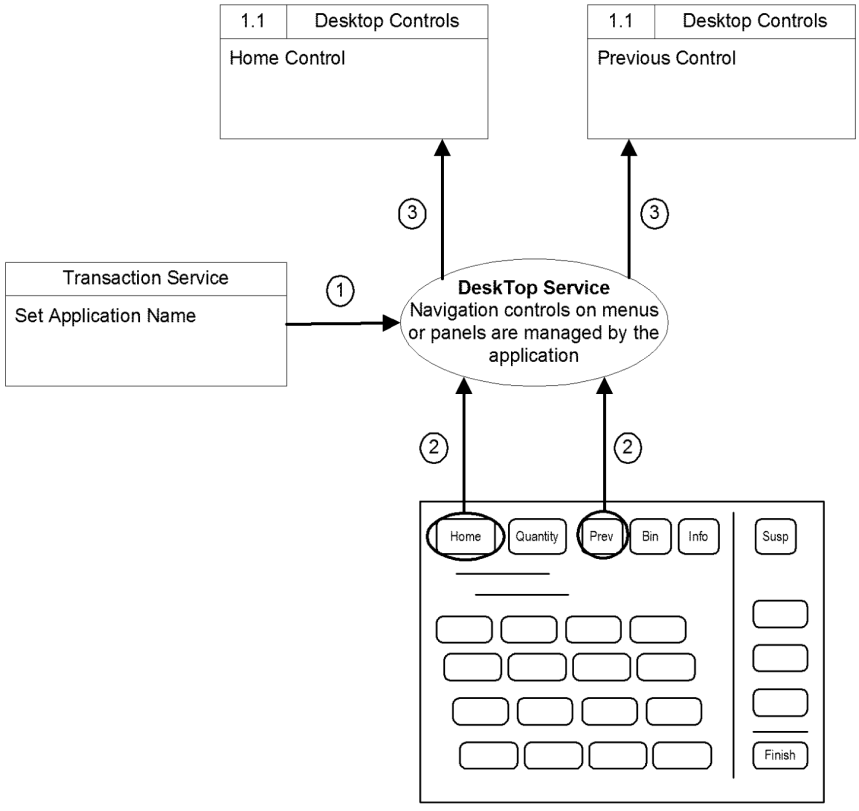


Figure 43 – External to Internal Mapping for Initiating Navigation Controls

FUJ00171948

5.3.3.9.3.1.2 Terminating Navigation Control

Unless an application indicates that it is running, as described in the previous section, the Desktop Service will manage the navigation of Home and Previous controls on all menus and panels displayed on the screen. In this case, the Desktop Service always returns to a predetermined menu when the user selects one of the navigation controls. Transaction Service also relinquishes control by clearing the application name held within the Desktop Service. This process is illustrated in Figure 44.

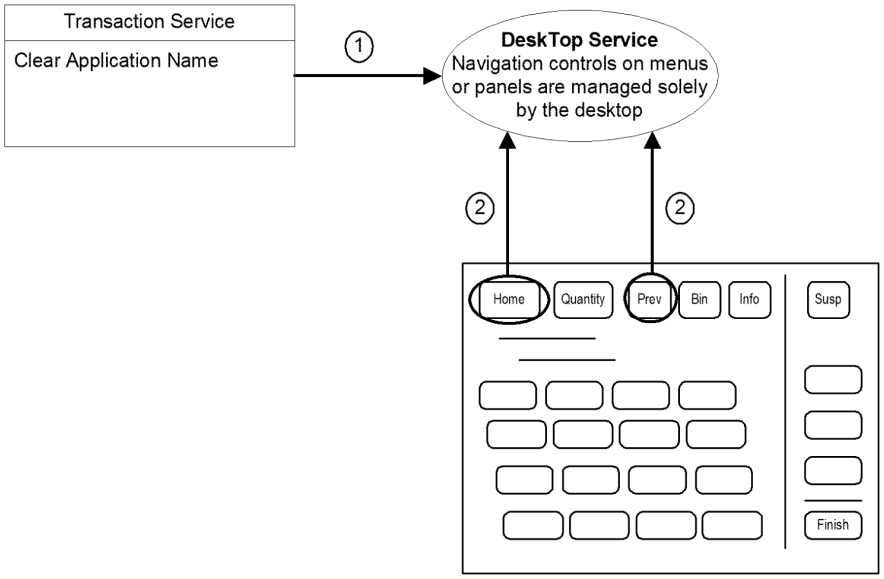


Figure 44 – External to Internal Mapping for Terminating Navigation Controls

FUJ00171948

5.3.3.9.3.2 Computational Model

The computational model for handling navigation of the Home control is given in Figure 45. Transaction Service sets the application name on the Desktop Service during a transfer out and an existing reversal session to handle the exceptions outlined in the diagram.

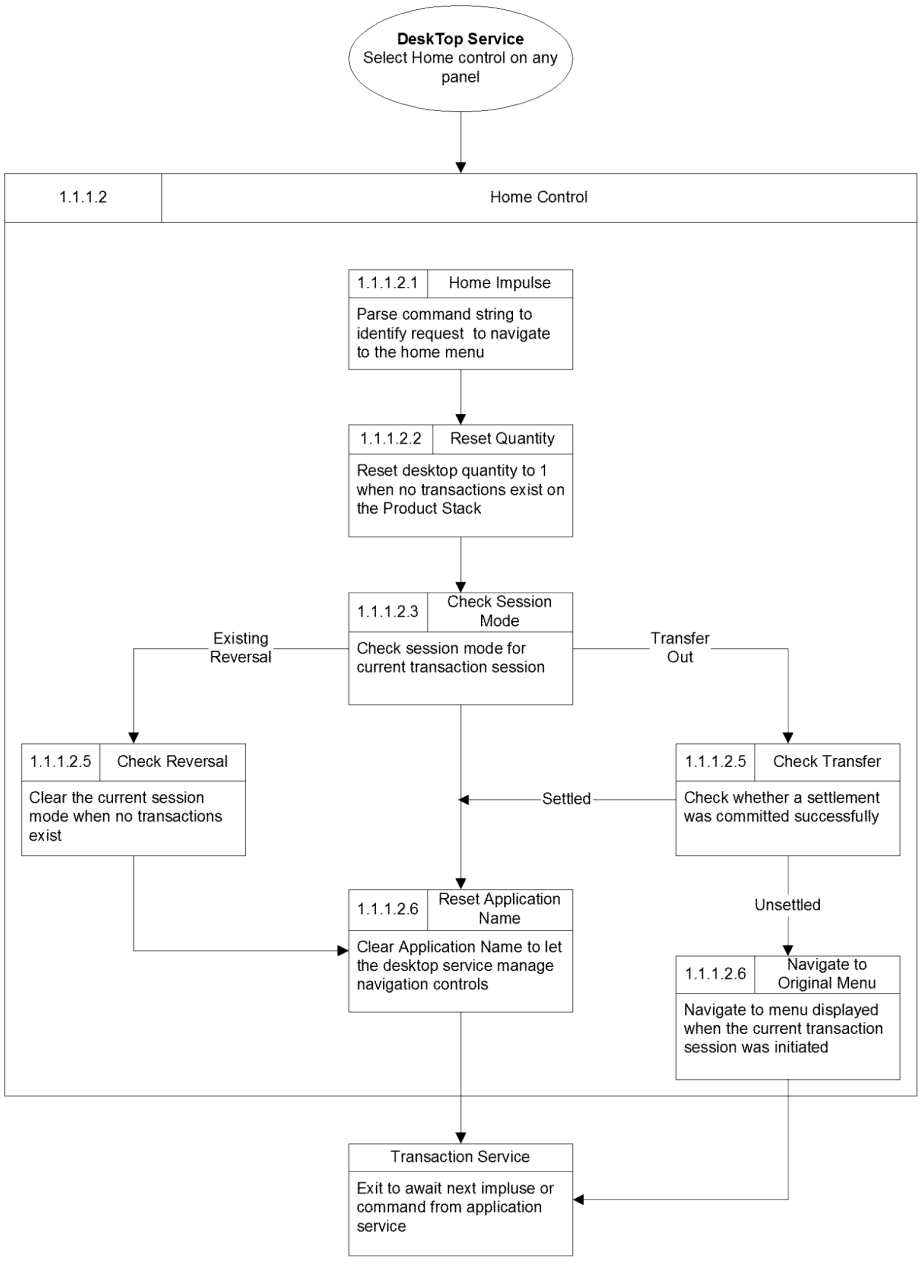


Figure 45 – Computational Model for Navigation of Home Control

FUJ00171948

5.3.3.9.4 Quantity Control

5.3.3.9.4.1 External to Internal Mapping

The Quantity control on a Transaction menu enables the user to set the number of items transacted during the current transaction session. If shopping mode is in progress the Quantity control contains the 'Shopping' caption and a Shopping panel is displayed during a transaction as described in section 5.3.3.8.8. Selection of Quantity control displays the Quantity panel and overrides shopping mode. See Figure 46.

A Quantity panel consists of a numeric keypad panel with an options sub-panel and a text box for entering a numeric value. The Quantity control on the Transactions menu is updated with the new value and extracted for subsequent transactions. Each new item selected is transacted in multiple units unless the quantity is outside the permitted range given by EPOSS Products, in Reference Data. The computational model is described and illustrated in section 5.3.3.8.4.

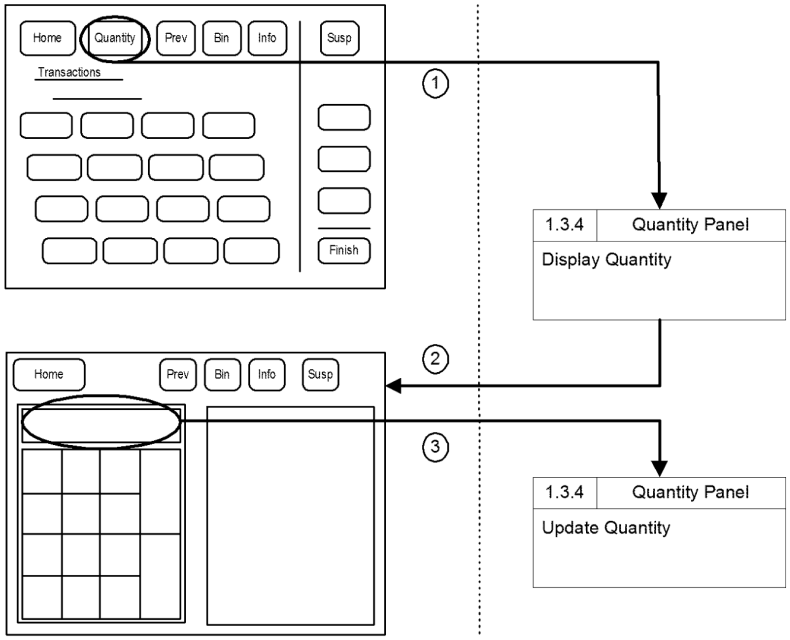


Figure 46 – External to Internal Mapping of Quantity Control

FUJ00171948

5.3.3.10 Desktop Menus

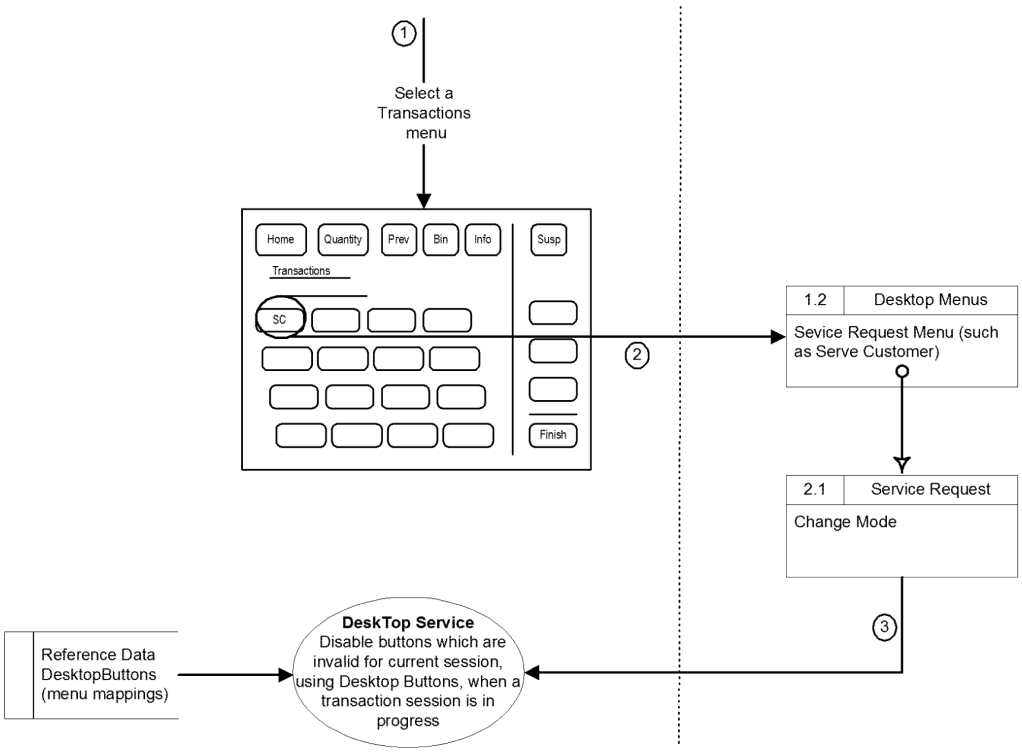
5.3.3.10.1 Service Request Menus

5.3.3.10.1.1 External to Internal Mapping

The user initiates the Transaction Service when the type of business transaction, known as the session mode, is selected from the Transaction menu or one of its submenus. The Transaction menu is displayed when the user chooses the Transaction button on the home menu.

A transaction session is initiated when a session mode is established with the Desktop Service. During initialisation the information held by Desktop Buttons in Reference Data is passed to the Desktop Service. This enables the Desktop Service to disable menu buttons that are not applicable to the current session mode, set by Change Mode, as shown in Figure 47.

Figure 47 – External to Internal Mapping of Service Request Menus



FUJ00171948

5.3.3.10.1.2 *Computational Model*

Each session mode is associated with a specific button on the Transactions menu or one of its submenus. Serve Customer, Housekeeping, Bulk Input (Recovery), Non-Account Data and Parcel Traffic modes can be selected directly from the Transactions menu. Otherwise the Transfers, Remittance, Reversals and Revaluation buttons on the Transaction menu are used to navigate to one of the other session modes. See Figure 48.

The computational model demonstrates that each service request menu is managed by one process, namely Change Mode. Depending on the session mode selected, Change Mode will navigate the user to the next sequence of menus and panels until a transaction session can take place. See section 5.3.4 for a description and illustration of the computational model for Change Mode.

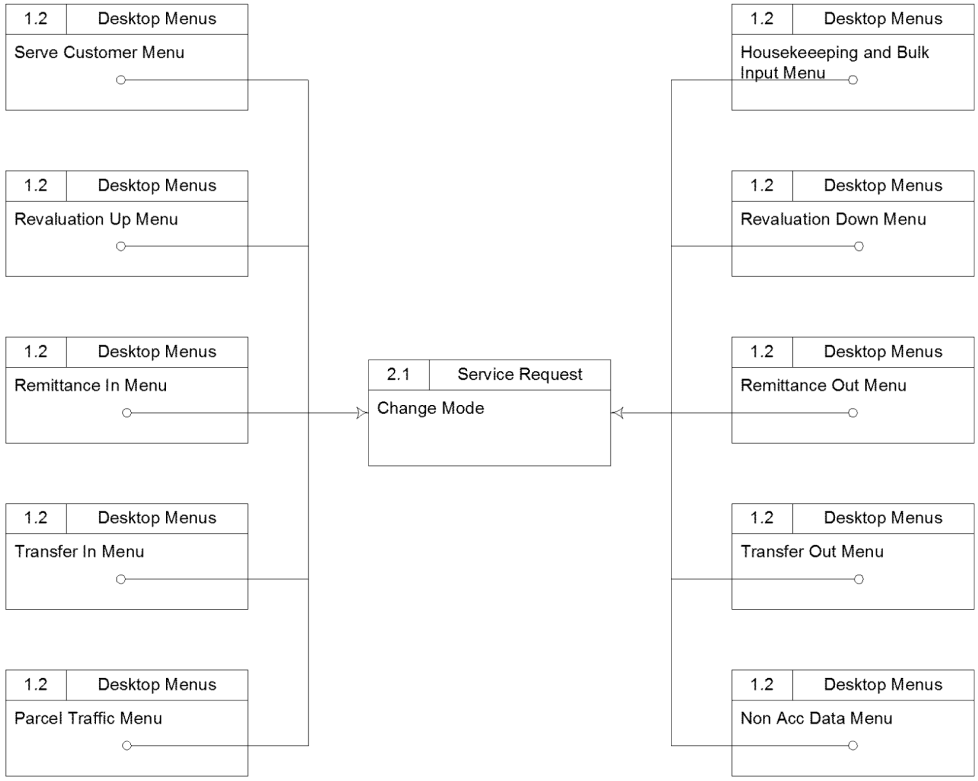


Figure 48 – Computational Model for Service Request Menus

FUJ00171948

5.3.3.10.2 Product Menus

5.3.3.10.2.1 External to Internal Mapping

When the user has established a session mode, such as Serve Customer, a product menu is displayed. A product menu contains sale items, represented by product buttons, for transacting specific products during a transaction session. Other buttons may be used to display another menu of products or a panel such as PLU List or PLU No, allowing one or more products to be chosen from a pick list or by its identity.

Product buttons contain a caption containing the product name, the keystroke that can be used for its selection from the keyboard and a graphical icon for user recognition. All core products are represented on a product menu with a product button. Non-core products may also be represented but only if the product can be sold at the given outlet.

Products designated on the desktop system by a specific button can be transacted by selecting the button from the appropriate product menu. Selection of a product using this method will result in the execution of a sales transaction for the number of items given by the current value on the quantity control.

Selection of a product, such as a first class stamp from the Serve Customer menu, generates an impulse, known as Sell Product, for Transaction Session. Sell Product creates a transaction for the given product and if successful, adds an icon representing the product to the bottom of the Product Stack as shown in Figure 49.

The Desktop Service will recalculate the session balance by adding the sale value of the item(s) transacted. The result is used to update the running total displayed on the Finish button caption.

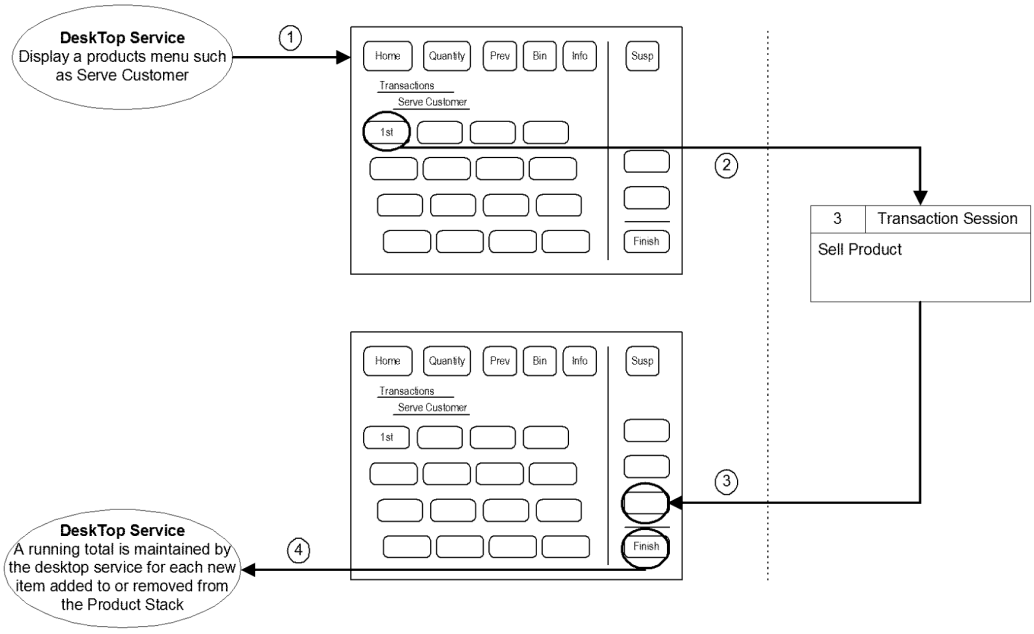


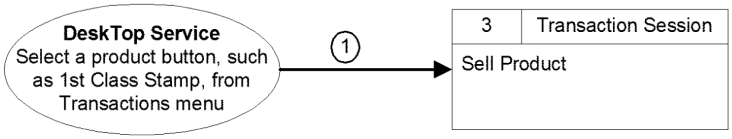
Figure 49 - External to Internal Mapping of Product Menus

FUJ00171948

5.3.3.10.2.2 Computational Model

The computational model for handling product buttons on product menus is given in Figure 50. Transaction Session receives an impulse, known as Sell Product, when the user selects a product, such as a first class stamp, from the Serve Customer menu.

Figure 50 – Computational Model for Product Menus



FUJ00171948

5.3.3.10.3 Settlement Menu

5.3.3.10.3.1 External to Internal Mapping

The end of the selling phase for a customer session is determined by selection of the Finish button displaying the customer session total. If the session balance is non-zero, the user will be offered various methods of payment on a settlement menu, illustrated in Figure 52. Selecting a method of payment will cause the Transaction Service to recalculate the session balance. When the session balance is zero all transactions on the Product Stack is committed to message store.

When the user has completed the sale of products during a session mode, such as Serve Customer, a settlement menu is displayed. A settlement menu contains payment items, represented by product buttons, for transacting specific methods of payment during a transaction session. Other buttons may be used to display another menu of payments.

A product button on a settlement menu appears and operates in the same way as a product button on a product menu, as described in section 5.3.3.10.2. A payment item is used to counterbalance the sale items transacted and therefore has the opposite affect on the session balance.

The session balance of a transaction session is always displayed in the form of a settlement to be paid in to the Post Office or paid out to the customer. A positive value indicates that a payment is outstanding and a debt is owed to the Post Office otherwise the customer is in credit. When the settlement function is invoked during a customer session, Transaction Service will present the user with several methods of payment given in the table below.

Settlement Product	Description
Cash	All or part of the current total value of the session is being paid with cash. The amount paid is input by the user
Fast Cash	The current total value of the session is being paid in full with cash.
Cheque	All or part of the current total value of the session is being paid with a cheque. The amount paid is input by the user
Fast Cheque	The total value of the session is being paid in full with a cheque.
Other Payments	Some payments, such as saving stamps, can only be used against related sale items. The amount paid is input by the user

Figure 51 – Table of Types of Settlement Product on Settlement Menu

FUJ00171948

During the selling phase, selection of the Finish button invokes the payment phase known as Settle Transactions in Transaction Session. For customer, new reversal and bulk input (recovery) sessions the settlement menu is displayed. Other session modes, described later, are automatically settled, via Sell Product, using a known settlement product reserved for the session mode. A settlement product adopts the settlement value and is transacted with the outstanding session balance.

Selection of a method of payment, such as Cash from the Settlement menu, generates an impulse, known as Sell Product, in Transaction Session. Sell Product creates a transaction for the given payment and adds an icon representing the payment to the bottom of the Product Stack as shown in section 5.3.3.10.2. The session balance is then determined by Settle Transactions from the Desktop Service.

Selection of a method of payment, such as Fast Cash from the Settlement menu, also invokes Sell Product in Transaction Session. However this method of payment is automatically settled because Fast Cash is a settlement product. A settlement product can adopt the settlement value as described above.

The Desktop Service will recalculate the session balance by subtracting the sale value of the item(s) transacted. The result is used to update the running total displayed on the Finish button caption.

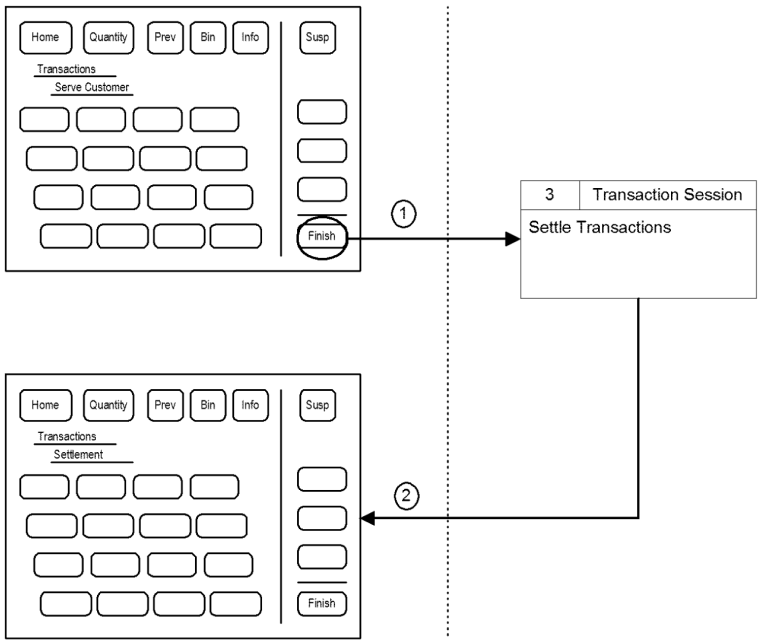


Figure 52 - External to Internal Mapping of Settlement Menus

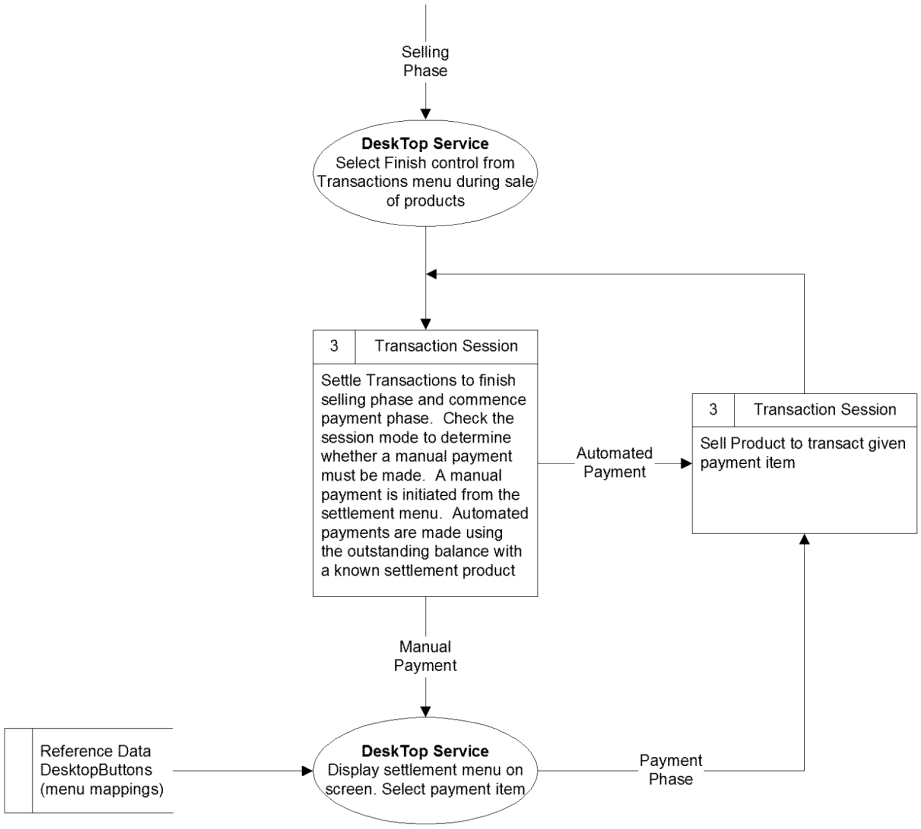
FUJ00171948

5.3.3.10.3.2 Computational Model

The computational model for handling method of payment buttons on product menus is given in Figure 53. Transaction Session receives an impulse, known as Sell Product, when the user selects a payment, such as Cash, from the Settlement menu. The selling phase is finished and the payment phase started when the user decides to select the Finish control on a product menu.

The settlement menu is displayed, forcing the user to make manual payments during Serve Customer, New Reversal or Bulk Input (Recovery) session modes. Otherwise Sell Product is used to transact an automated payment against a known settlement product associated with the current session mode.

Figure 53 – Computational Model for Settlement Menu



* An unbalanced transaction session is assumed during settlement

FUJ00171948

5.3.4 Service Request

5.3.4.1 Change Mode

5.3.4.1.1 Setting Session Modes

Business transactions are controlled by session modes. A session mode must be instantiated before a transaction can take place. Each session mode will impose certain limitations on the desktop system. For example, if the current session mode is Serve Customer the user is prevented from accessing and transacting stock transfers and remittances. The range of session modes is illustrated in Figure 54.

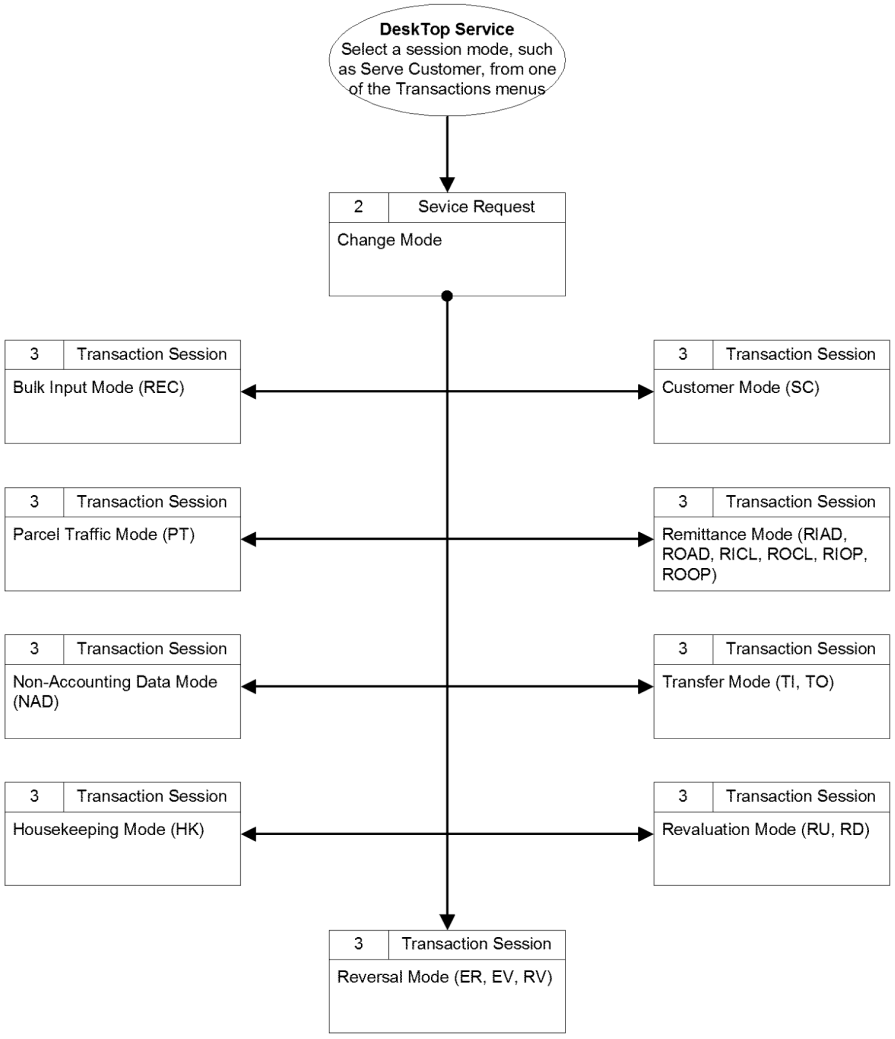


Figure 54 – Setting a Session Mode for Transaction Session

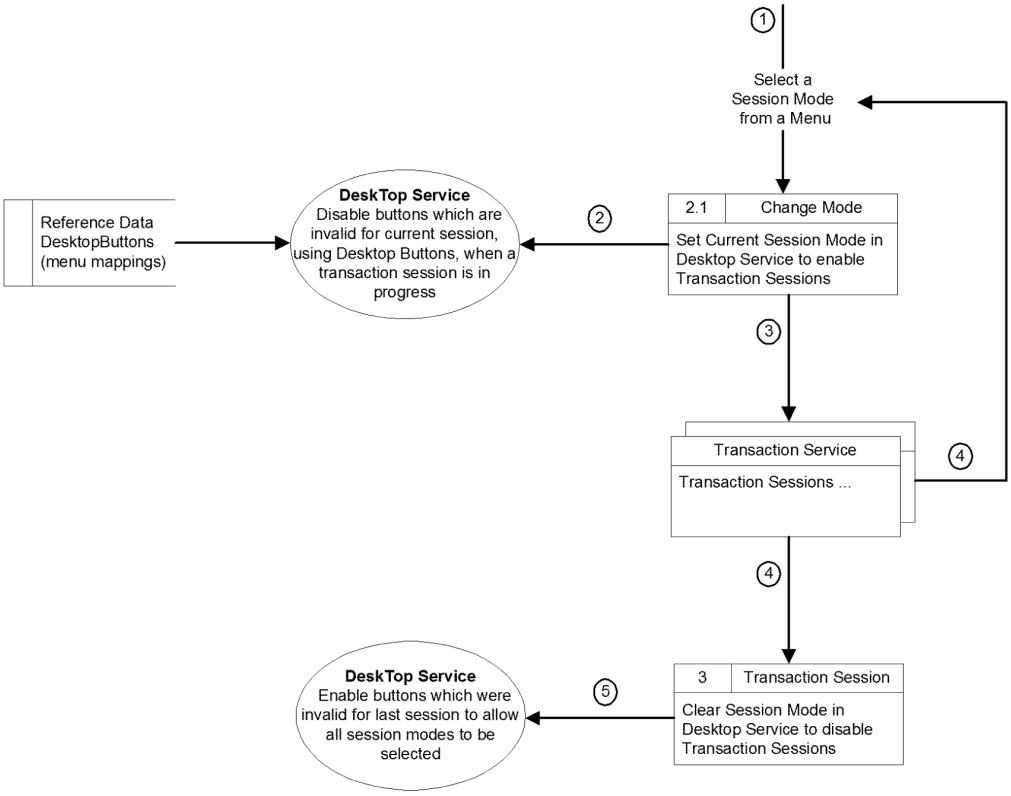
FUJ00171948

5.3.4.1.2 Initiating Transaction Sessions

A transaction session can be initiated when the session mode has been established with the Desktop Service. Figure 55 shows the effect of setting or clearing the current session mode in the Desktop Service.

Setting and clearing the session mode disables and enables desktop buttons on menus as directed by Desktop Buttons in Reference Data. In general, other session modes are disabled when a session mode is initiated.

Figure 55 – Initiating Transaction Sessions using Session Modes



FUJ00171948

5.3.4.1.3 Interaction between Remittance Sessions

A remittance session involves the movement of cash and stock items between an outlet and one of a number of external sources or destinations. The range of products that can be remitted is dependent on the outlet and the external source or destination defined by Reference Data. The external sources and destinations supported by the system are listed below.

- Supplies Division
- Auto Distribution Centre
- Client
- Other Post Office

External movements of cash and stock items involve remittance in and remittance out operations. The external source of a remittance session is identified by the session mode. For example the RIAD and ROAD session denote a remittance between the Auto Distribution Centre and the outlet.

A remittance in session increases the level of stock within the stock unit assigned to the current user. A remittance out session has the opposite effect. Transfer In and Transfer Out sessions affect stock levels of stock units in the same way.

The process for identifying items and quantities to be remitted are the same as a Serve Customer session. Unlike transfer sessions, described in section 5.3.4.1.4, there is no interaction between remittance in and remittance out operations.

Remittances are supported by manual procedures requiring reports of cash and stock items sent and received by an outlet. A report is generated giving the value of the cash and stock items remitted in or out at the outlet against the stock unit assigned to the user during the remittance session. Discrepancies between the system and the manual process are recorded separately via a housekeeping session at the outlet.

FUJ00171948

5.3.4.1.4 Interaction between Transfer Sessions

A stock transfer is an internal movement of cash and stock items between stock units assigned to the outlet. Unlike remittance sessions, described in the previous section, there is interaction between transfer in and transfer out operations, illustrated in Figure 56. A transfer out session must be transacted and completed successfully before a transfer in operation can be invoked on the same stock unit.

Before a transfer out session can be started the user must identify the receiving stock unit from a pick list of available stock units assigned to the outlet. This list includes only those stock units working in the same Cash Account Period as the user assigned stock unit. A menu of products is displayed, illustrated in 5.3.4.1.5.2, enabling the user to start the stock transfer session.

When the transfer out session is complete details of the session are recorded and added to the Stock Transfers List attached to the stock unit selected by the user. The Stock Transfers List contains session details of transfer out operations successfully completed on the associated stock unit. The Stock Transfers List is utilised by transfer in sessions.

Before a transfer in session can be started the user must identify the transfer out session from a pick list of completed transfers on the user assigned stock unit at the outlet. This list contains details of the originating stock unit, transfer session identifier and the total value of stock items transacted.

The user will be able to select and preview, but not change, the contents of a transferred item or print the details of the item. The transfer in session can either be abandoned or accepted. Acceptance will invoke a transaction session to transfer in each item transacted during the transfer out operation into message store. The sale value of the original transaction is negated to generate a compensating transaction.

When the transfer in session is complete details of the session are removed from the Stock Transfers List of the stock unit assigned to the user. The user must reselect the transfer in session mode to transfer another transaction session from the Stock Transfer List into the user assigned stock unit.

FUJ00171948

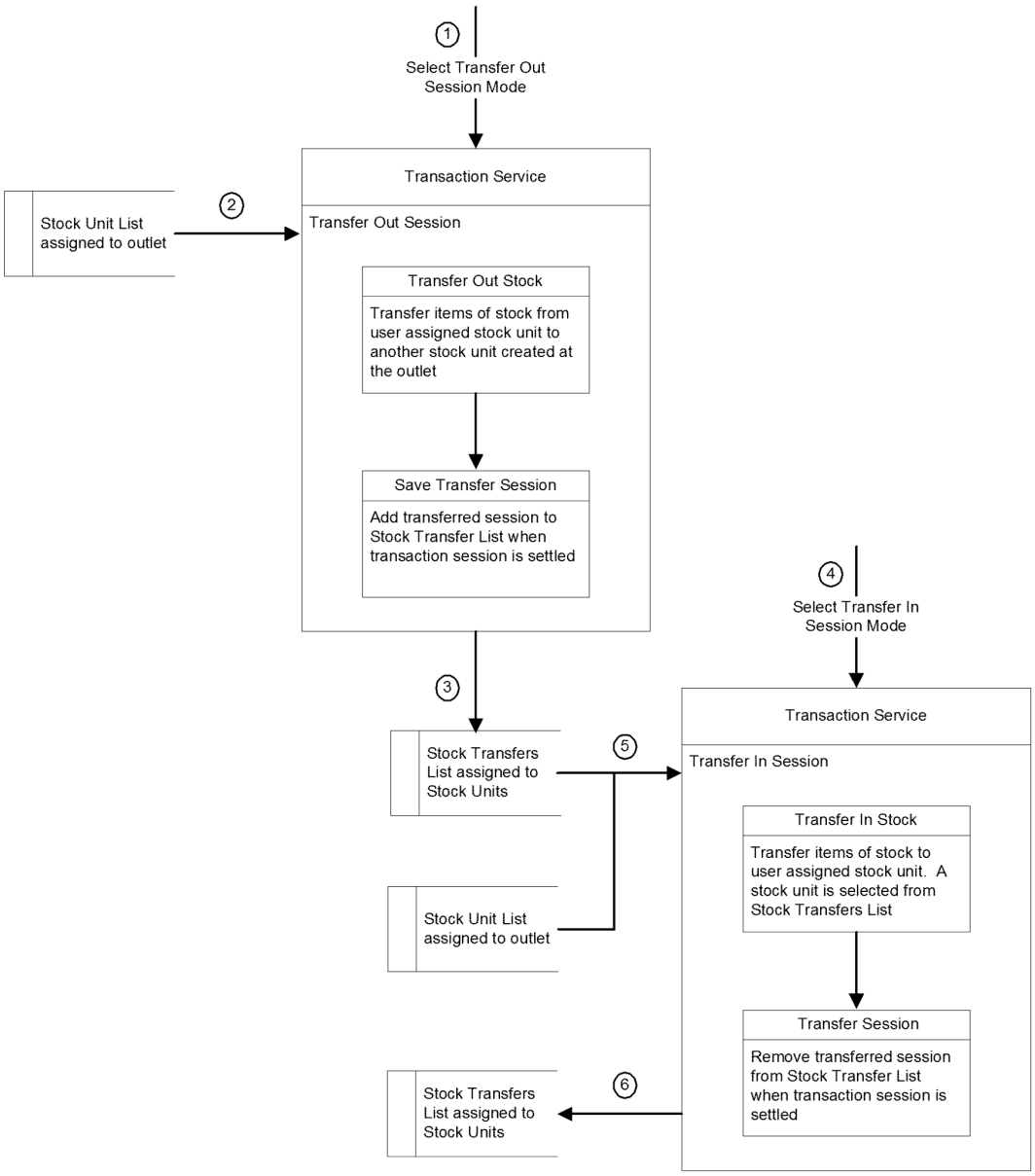


Figure 56 – Interaction between Change Mode and Transfer Sessions

FUJ00171948

5.3.4.1.5 External to Internal Mappings

5.3.4.1.5.1 Typical Session

Each session mode is associated with a specific button on a Service Request menu as described in section 5.3.3.10.1.2. Each service request menu is managed by one process, namely Change Mode. Depending on the session mode selected. Change Mode will navigate the user to the next sequence of menus and panels until a transaction session can take place, as shown by a typical session mode, namely Serve Customer, given in Figure 57.

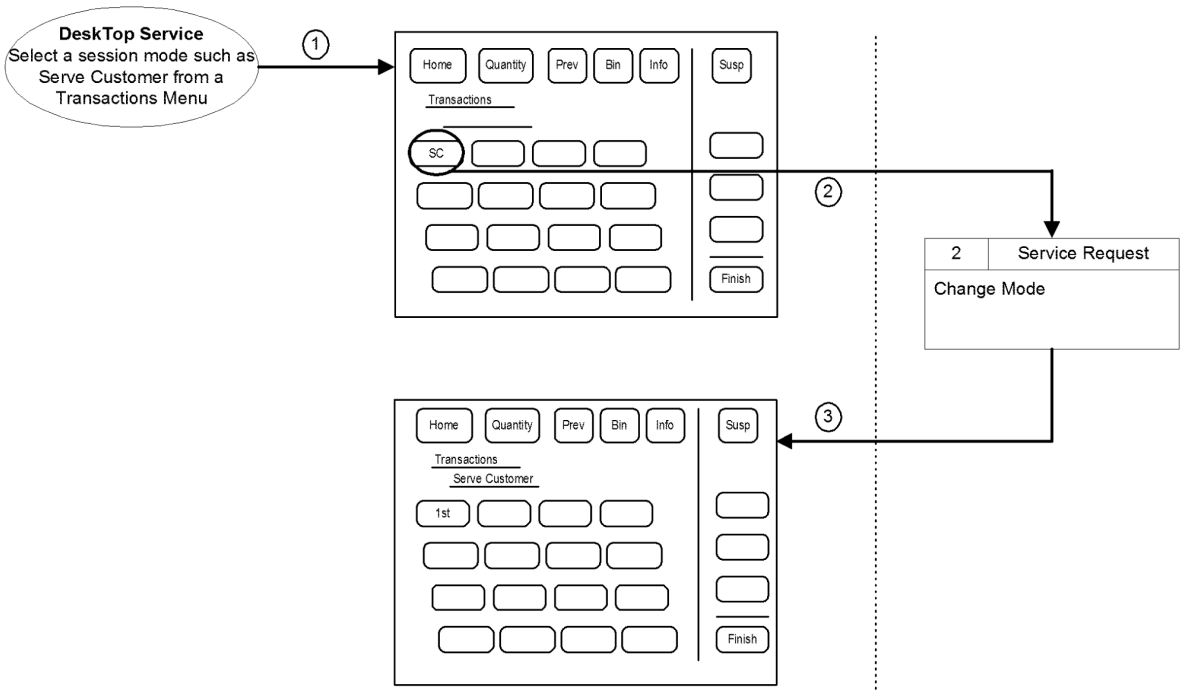


Figure 57 - External to Internal Mapping of Change Mode for Customer Session

FUJ00171948

5.3.4.1.5.2 *Transfer Out Session*

When the Transfer Out button is selected from the Transfers menu the user is presented with a pick list of available stock units assigned to the outlet. This list includes only those stock units working in the same Cash Account Period as the user assigned stock unit but excludes the user assigned stock unit.

Selecting a stock unit displays a menu of products enabling the user to start the stock transfer session. A product is transacted in the same manner as a Serve Customer session, i.e. via Sell Product in Transaction Session. See Figure 58.

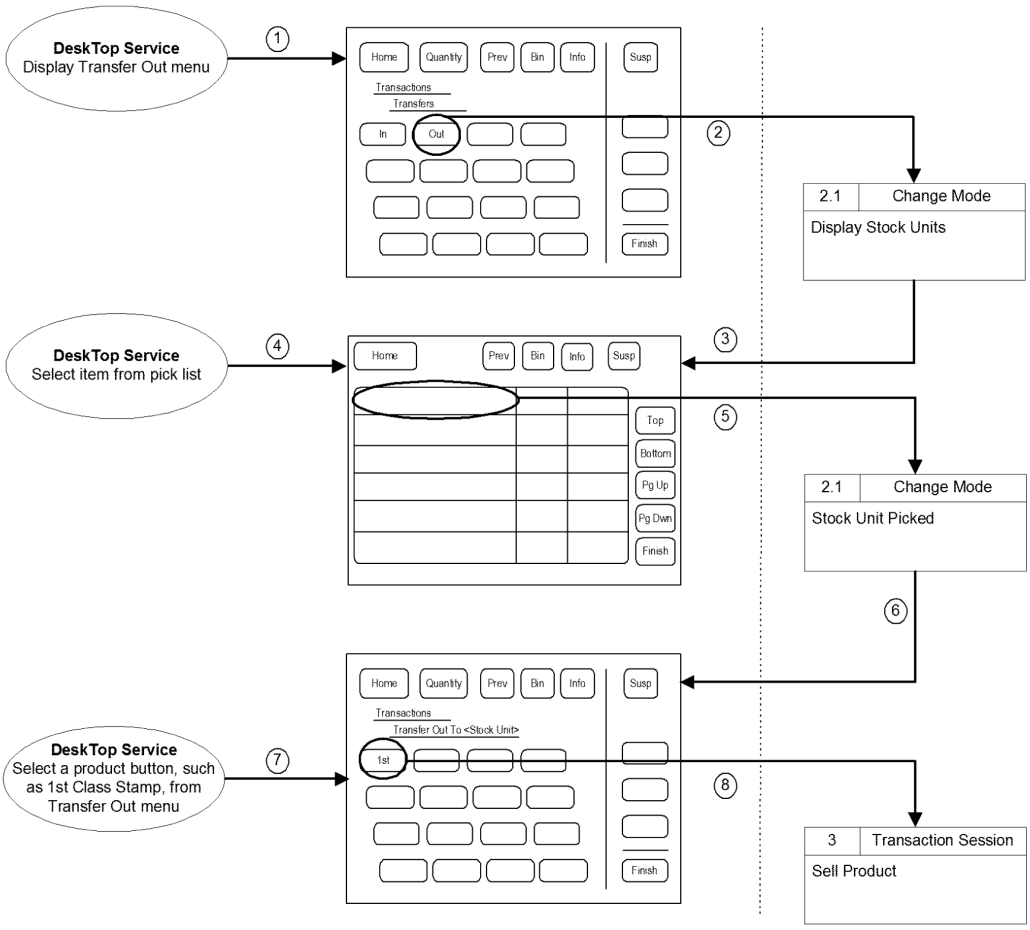


Figure 58 - External to Internal Mapping of Change Mode for Transfer Out Session

FUJ00171948

5.3.4.1.5.3 *Transfer In Session*

When the Transfer In button is selected from the Transfers menu the user is presented with a pick list of transfer out sessions involving the stock unit assigned to the user at the outlet. This list contains details of the originating stock unit, transfer session identifier and the total value of stock items transacted.

Selecting a transfer out session displays a panel summarising the details of the session. Controls are also provided on the panel enabling the user to preview or print further details of the session selected. Transactions generated by this session can be reviewed before the transfer in operation is invoked. Acceptance starts the stock transfer session, which generates a compensating transaction for each transaction in the original transfer out session, via Sell Product in Transaction Session. See Figure 59.

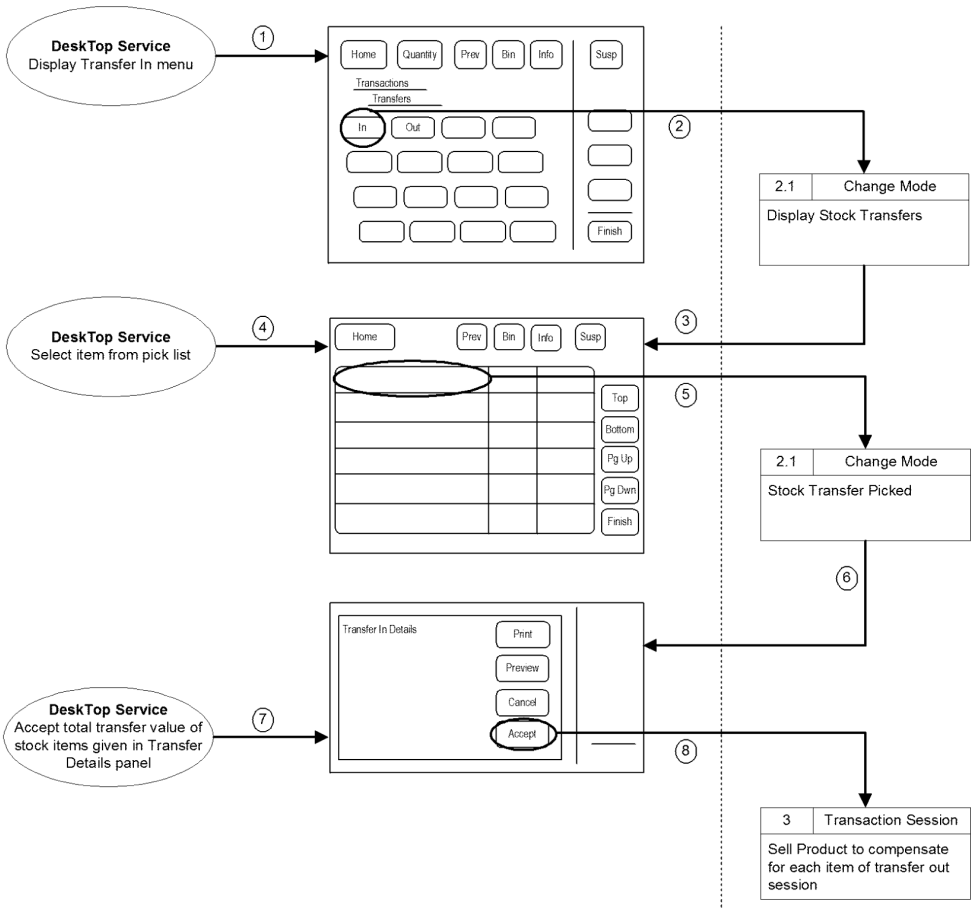


Figure 59- External to Internal Mapping of Change Mode for Transfer In Session

FUJ00171948

5.3.4.1.6 Computational Models

5.3.4.1.6.1 New Session Mode

The computational model for changing the current session mode, as directed by the user from a service request menu, is given in Figure 60. The range of session modes and the effect on other modes when a new mode is selected is described and illustrated in sections 5.3.4.1.1 and 5.3.4.1.2, respectively.

Change Mode receives an impulse, known as Change Mode, when the user selects a session mode, such as Serve Customer, from a Transactions menu. In general, when a session mode is instantiated other session modes are disabled, preventing conflicts between incompatible session modes.

The internal transfer of stock items, described and illustrated in section 5.3.4.1.4, involves inter process communication between transfer in and transfer out operations. A transfer in session is dependent on pending transfer out sessions to the user assigned stock unit. Change Mode establishes the stock units and transfer session required for the internal transfer of stock items to or from the user assigned stock unit.

A service request for a Transfer Out session enters the Transfer-Out Mode path, described in section 5.3.4.1.6.2. A service request for a Transfer In session enters the Transfer-In Mode path, described in section 5.3.4.1.6.3. In both cases, a panel is displayed prompting the user to select an item from a pick list before proceeding with the transfer session. When an item is selected from the pick list the path for a typical transaction session is taken.

A menu of products or panel of transactions for the new session mode is determined from the command string and displayed on the screen. This has the side effect of updating the current session mode to the requested session mode given in the command string. From this point onwards the current session mode is now the new session mode.

Discount buttons at CSR+ are no longer used and are disabled for all session modes. Customer discount buttons appear on their respective menus for the Serve Customer session mode only. Bulk Input discount buttons appear for the Recovery session mode only.

FUJ00171948

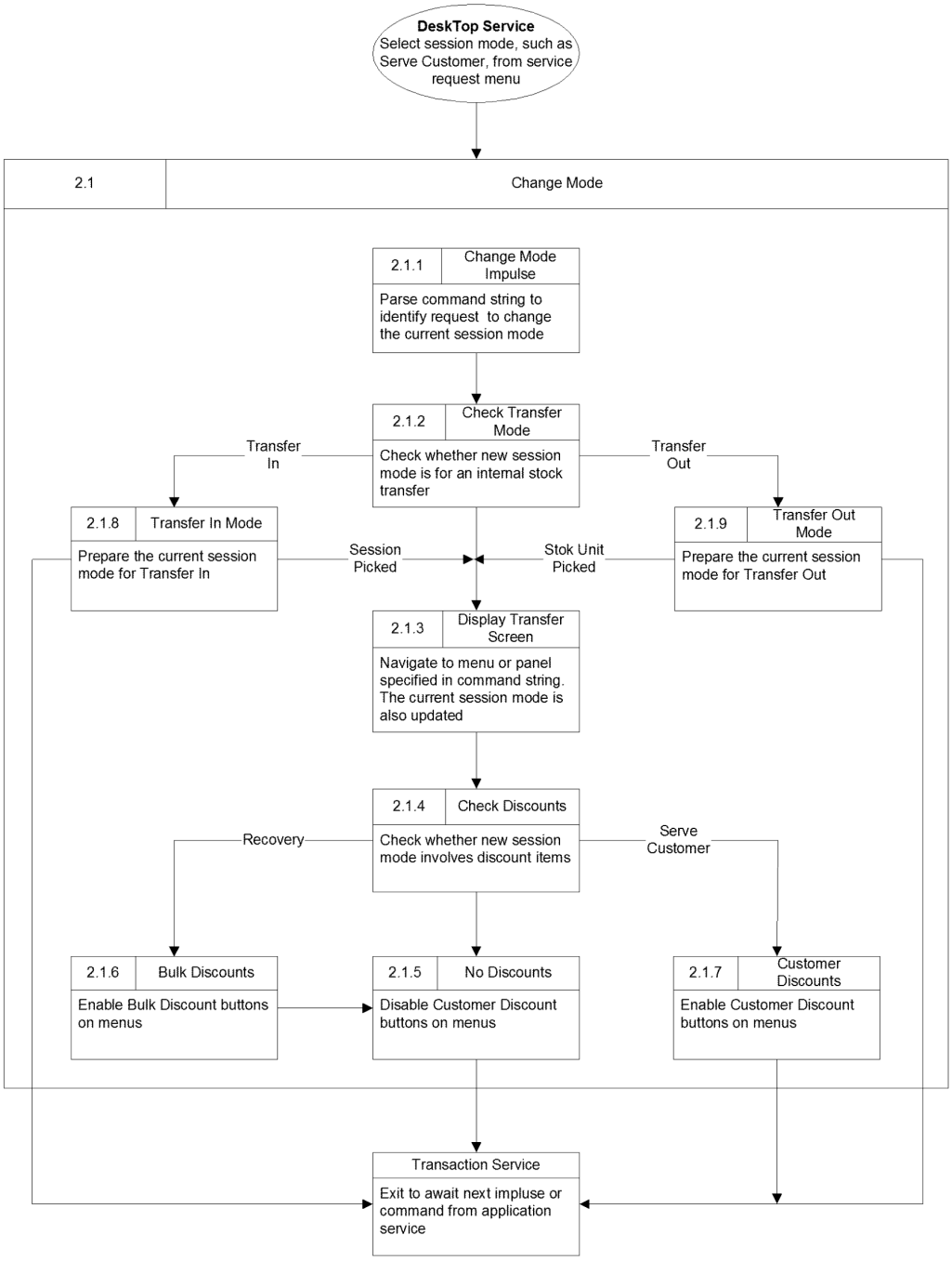


Figure 60 – Computational Model to Initiate a New Session with Change Mode

FUJ00171948

5.3.4.1.6.2 Transfer Out Mode

The computational model for instantiating the transfer out session mode, as directed by the user from the Transfers menu, is given below. Change Mode parses the command string and establishes a service request for a Transfer-Out session and enters the Transfer-Out Mode path, described in section 5.3.4.1.6.1.

A List of Outlet Stock Units, containing all stock units created by the administrator, is used to create a local collection of stock units. This collection includes only those stock units working in the same Cash Account Period as the user assigned stock unit but excludes the user assigned stock unit.

A pick list of available stock units assigned to the outlet is generated from the local collection of stock units and presented to the user via the Stock Unit panel. Selecting a stock unit establishes the target stock unit for transfer of stock items from user assigned stock. To process exceptions during a transfer out session the Application Name is set to manage navigation controls on product menus.

The name of the target stock unit is saved for display as part of the menu subtitle. A menu of products is displayed, with a menu subtitle stating the name of the target stock unit, enabling the user to start the stock transfer session. The session mode is then changed as described and illustrated in 5.3.4.1.6.1.

A product is transacted in the same manner as a Serve Customer session, i.e. via Sell Product in Transaction Session. When the transfer out session is complete details of the session are recorded and added to the Stock Transfers List attached to the stock unit selected by the user. See 5.3.4.1.4.

FUJ00171948

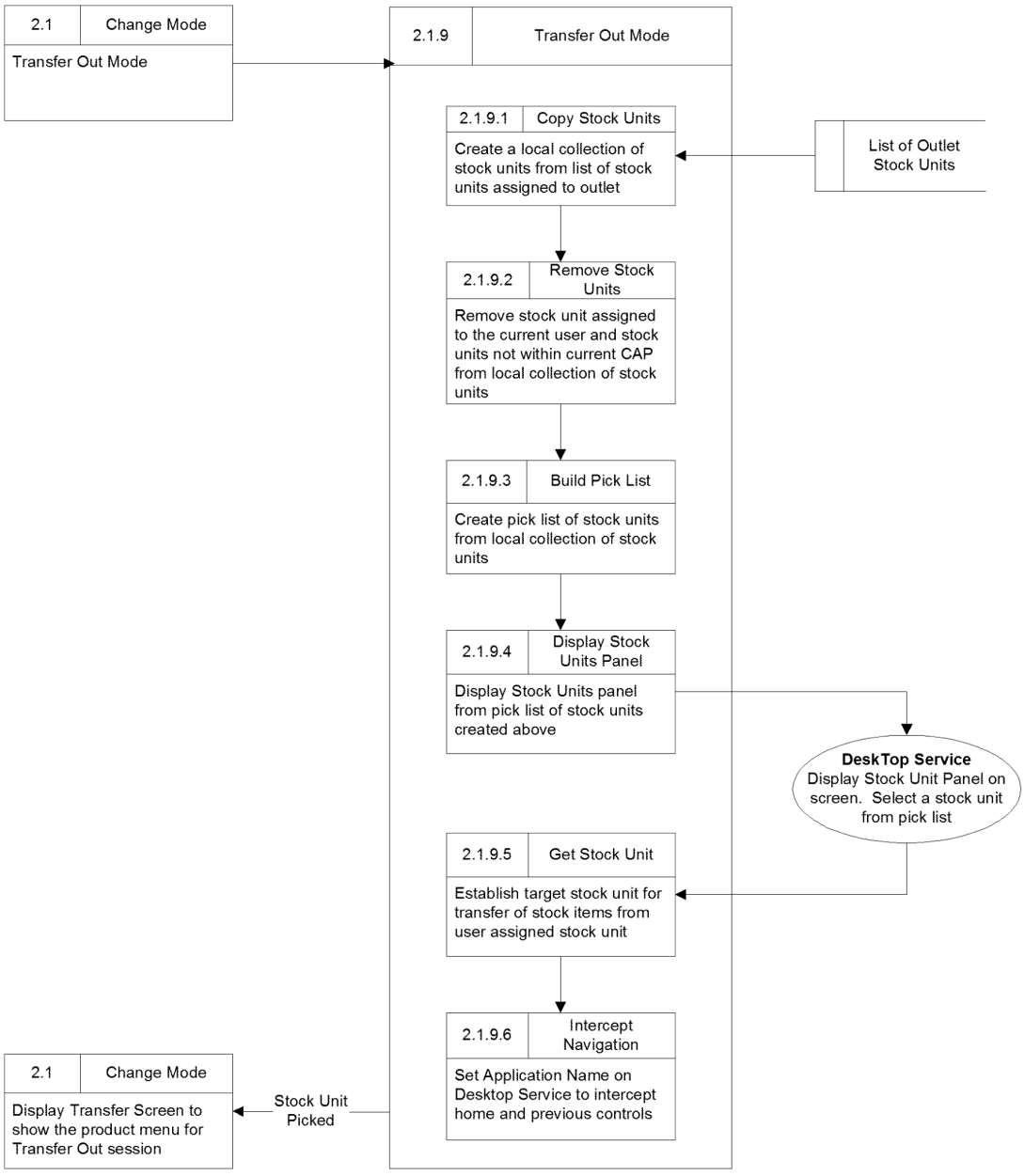


Figure 61 – Computational Model to Initiate Transfer Out with Change Mode

FUJ00171948

5.3.4.1.6.3 Transfer In Mode

The computational model for instantiating the transfer in session mode, as directed by the user from the Transfers menu, is given below. Change Mode parses the command string and establishes a service request for a Transfer-In session and enters the Transfer-In Mode path, described in section 5.3.4.1.6.1.

A List of Transfer Sessions, containing pending transfer sessions, successfully completed by transfer out operations at the outlet, is used to create a local collection of transfer sessions. This collection includes only those transfer sessions associated with the user assigned stock unit.

A pick list of transfer sessions attached to the user assigned stock unit is generated from the local collection of transfer sessions and presented to the user via the Transfer Sessions panel. Selecting a transfer session establishes the source stock unit for transfer of stock items to the user assigned stock unit.

The name of the source stock unit is saved as part of the subtitle for the Transfer In panel displayed next. A panel is displayed, with a menu subtitle stating the name of the source stock unit, enabling the user to start the stock transfer session. The transfer in session can either be abandoned or accepted. Acceptance will update the session mode as described and illustrated in section 5.3.4.1.6.1.

The user will be able to select and preview, print, accept or abandon the transfer in session. Acceptance will invoke a transaction session to transfer in each item transacted during the transfer out operation into message store. When the transfer in session is complete details of the session are removed from the Stock Transfers List of the stock unit assigned to the user. See 5.3.4.1.4.

FUJ00171948

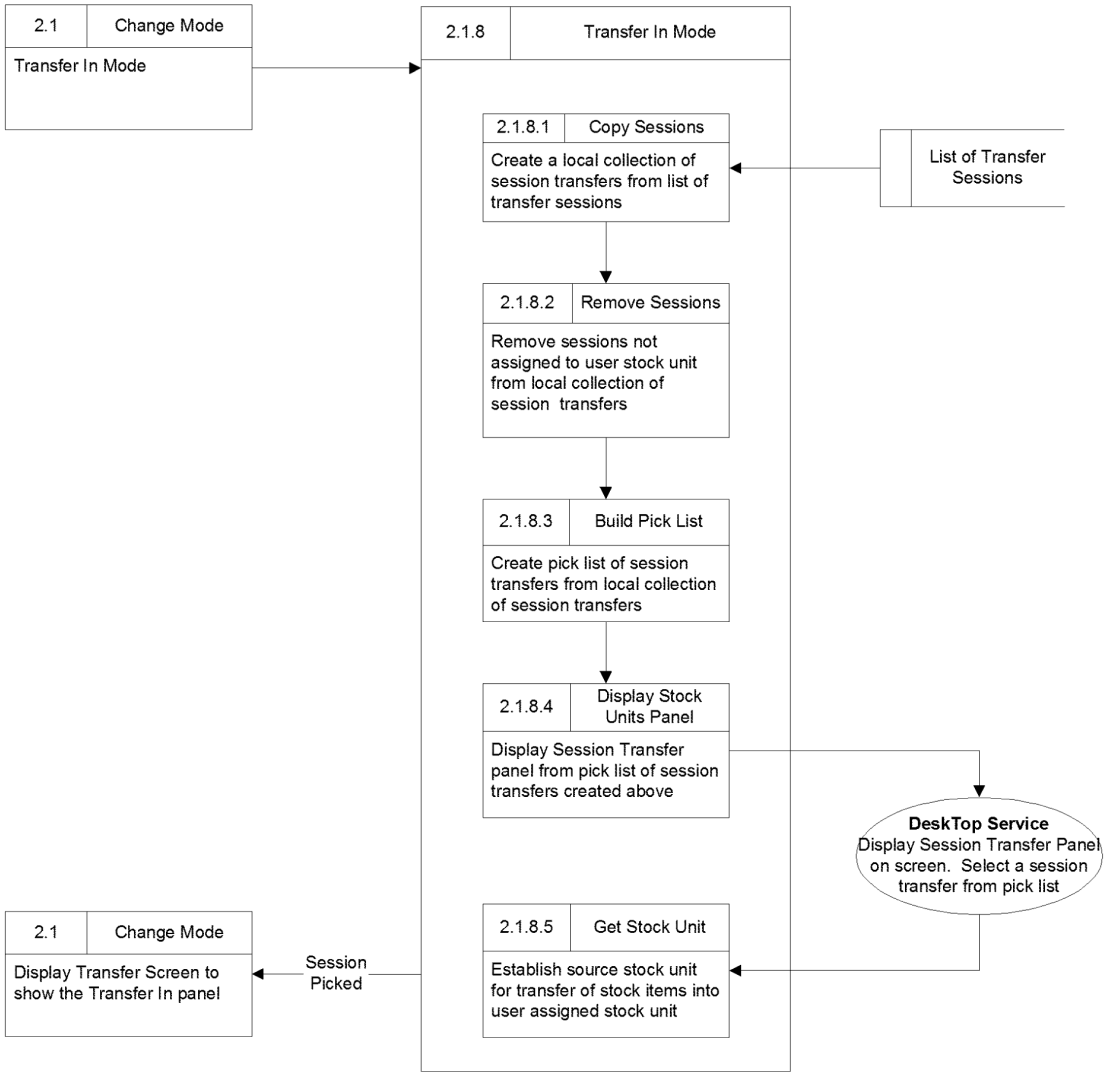


Figure 62 – Computational Model to Initiate Transfer In for Change Mode

FUJ00171948

5.3.5 Transaction Session

5.3.5.1 User Perspective

The user perspective of Transaction Session is illustrated in Figure 63. Before Transaction Session can be utilised the user must first select a session mode via a service request menu. A menu of products is displayed when a session mode such as Serve Customer is selected.

Selecting a product from the Serve Customer menu generates an impulse, known as Sell Product, for Transaction Session to create business transaction. Choosing a product from the PLU panel or a Pick List panel identifies a product number for Desktop Presentation to create a Sell Product command string.

Selecting the Home control navigates the user back to the home page irrespective of the status of a transaction session. When there is an outstanding transaction on the Product Stack the user will be unable to select another session mode until the current transaction session is completed or abandoned.

When the current transaction session is completed or abandoned the Product Stack is empty and the user can select another session mode after navigating back to the home page. A transaction session is abandoned by removing each item from the Product Stack.

FUJ00171948

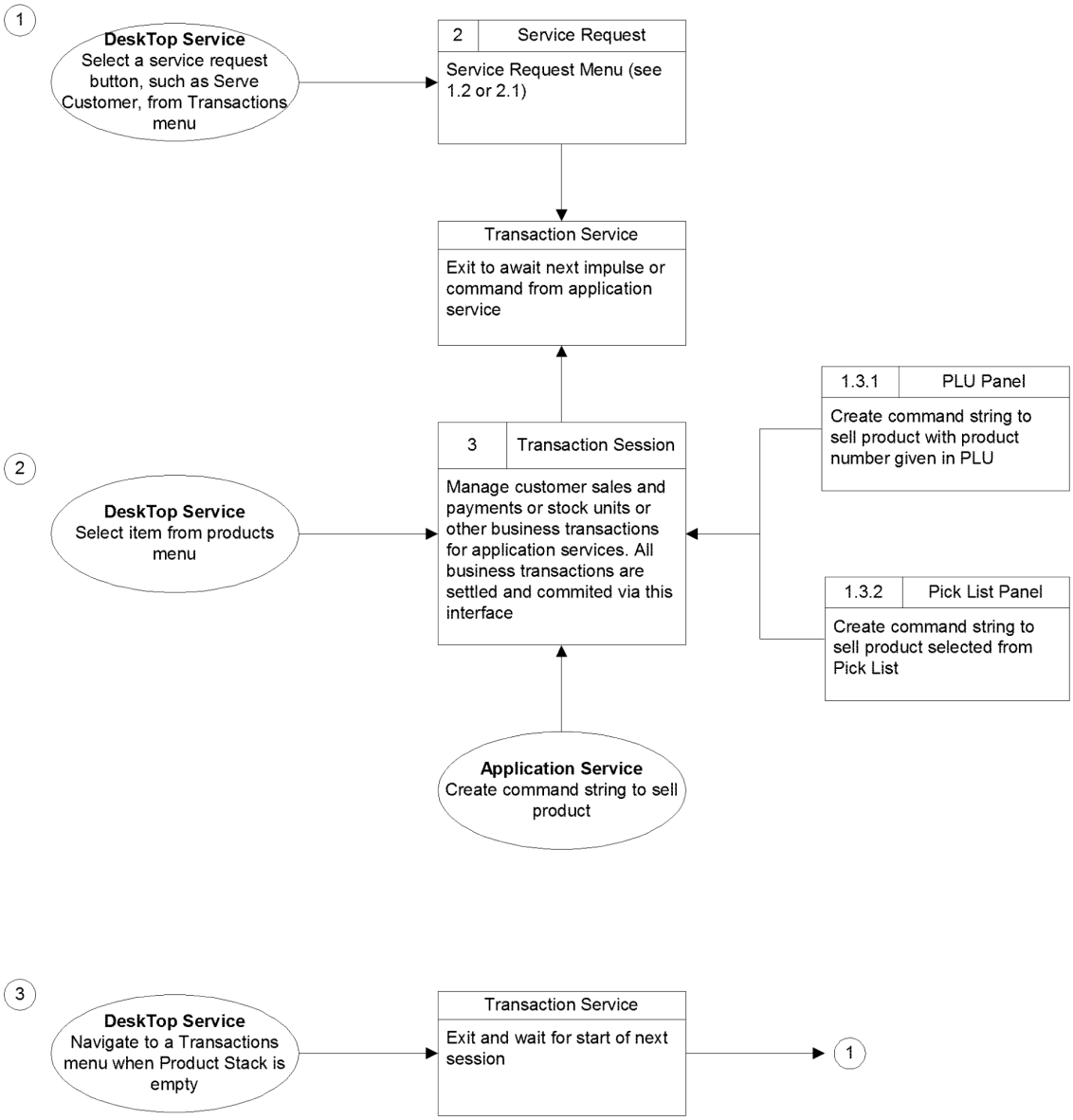


Figure 63 – User Perspective of Transaction Session

FUJ00171948

5.3.5.2 Development Perspective

The development perspective of Transaction Session is illustrated in Figure 64. Transaction Session is utilised by other application services in addition to the Desktop Service and Desktop Presentation.

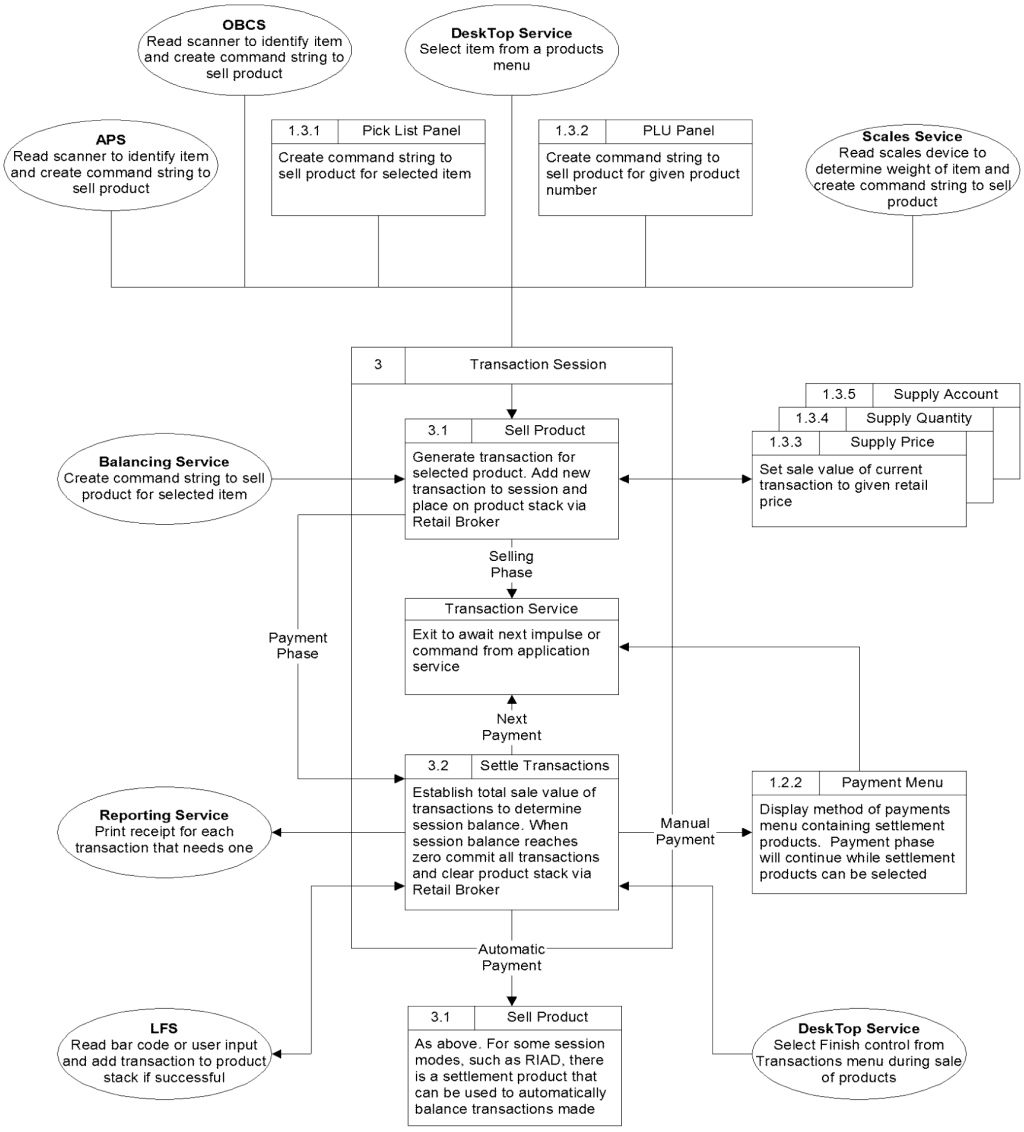


Figure 64 – Development Perspective of Transaction Session

FUJ00171948

5.3.5.3 Overview of a Transaction

An overview of the application components that comprise Transaction Session is given in Figure 65. This introduces the numbering scheme for logical components and reflects the hierarchy of subsequent subsections in this part of the document. Each subsection presents a logical abstract model of the underlying code.

Transaction Session is composed of two elements, known as Sell Product and Settle Transactions. Sell Product performs all business transactions on behalf of the user or another application service. Settle Transactions is invoked when the user selects the Finish control on the current menu.

Settle Transactions checks the current session balance calculated and saved on the Finish control by the Desktop Service. When the outstanding balance is zero, Settle Transactions settles the transaction session by committing each transaction to the message store, via Retail Broker. The Product Stack is also cleared during committal.

All transactions are generated by Sell Product irrespective of whether they involve sale or payment items. Sale items are transacted initially during the selling phase without involving Settle Transactions. When a payment item (also known as a settlement product) is transacted successfully the payment phase is started. Settle Transactions is invoked, by Sell Product, for each payment made.

Settle Transactions invokes Sell Product when the current session mode is associated with a known settlement product. In this case a settlement product is transacted with a sale value equal to the outstanding balance. Otherwise a method of payments menu is displayed to enable the user to make manual payments.

The basic elements of Sell Product and Settle Transactions are described in more detail by the computational models illustrated in section 5.3.5.4, 5.3.5.5 and 5.3.5.6. Figure 65 illustrates the interaction between Sell Product and Settle Transactions.

FUJ00171948

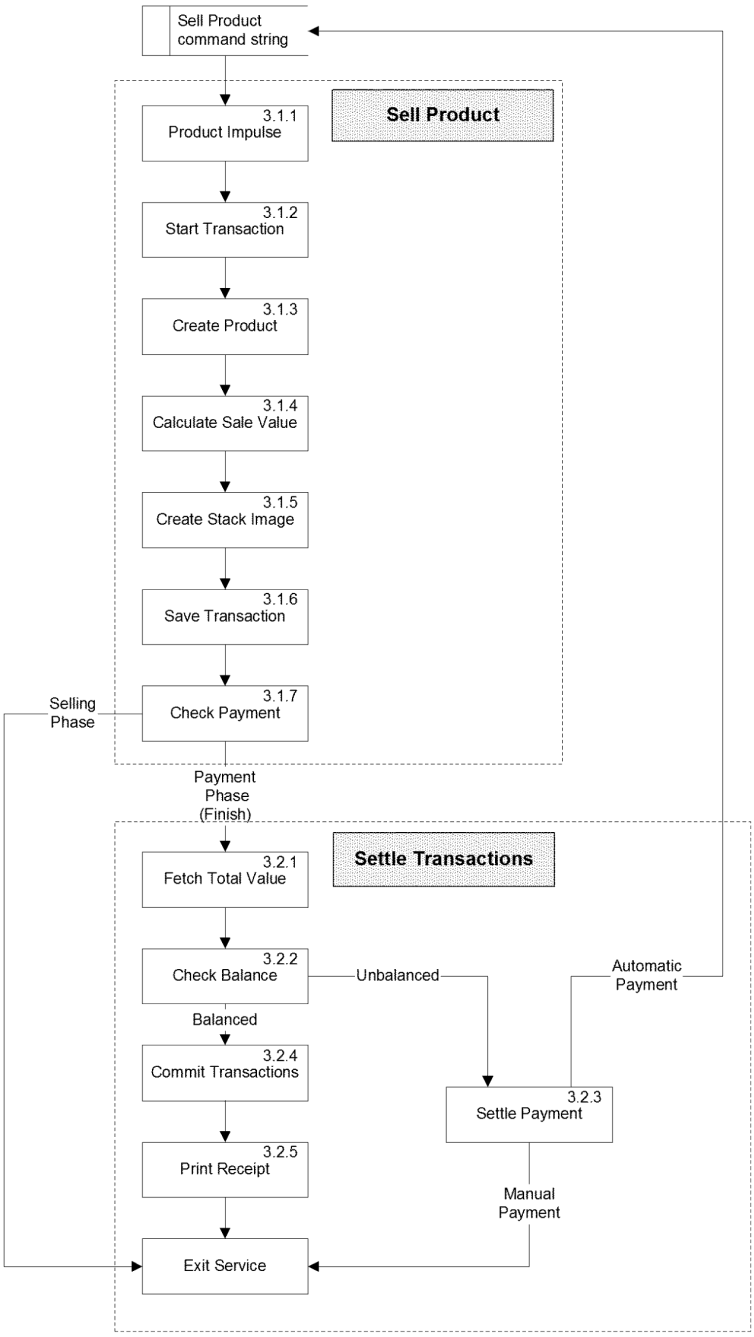


Figure 65 – Overview of a Transaction for a Sale or Payment Item

FUJ00171948

5.3.5.4 Sell Product

The computational model for the Sell Product component of Transaction Session is given in Figure 66. The business rules for generating a transaction are implemented within Create Product and Calculate Sale Value elements. These elements require further decomposition and are described in more detail in a later section of this document. For example, a transaction may involve the sale of more than one product or require additional data to be input by the user as dictated by product collections in Reference Data.

To estimate the time the system takes to process a transaction, Sell Product calls the Start Transaction method, provided by Retail Broker, before transacting the given product. Retail Broker records the start date and time of the new transaction. When the transaction has been fully processed, Sell Product passes the details of the transaction to Retail Broker via the WantToCommit method. Retail Broker records the completed date and time of the new transaction.

Create Product uses the product number of the item selected, given in the command string, to look up information contained in product collections of Reference Data. The attributes and properties of the product or service involved are extracted to create transaction data. This information is used to determine the transaction process, such as whether Supply Account, Supply Quantity and Supply Price are invoked.

Calculate Sale Value will invoke Supply Price, when an item has a default price, to prompt the user to accept or alter the given price. If an item is designated as open price Calculate Sale Value will invoke Supply Price to prompt the user to enter the price of an item that has no predetermined value. The sale value is calculated from Quantity * Unit Price – Discounts, where Quantity is the current value on the Quantity control and Discounts is the amount of discount calculated for a customer or member of staff.

Sell Product determines the bit map image of the product selected from the Product Stack collection in Reference Data. . When the transaction has been fully processed, the details of the transaction and the bit map image are passed to Retail Broker via the WantToCommit method. Retail Broker adds the bit map image of the product to the bottom of the Product Stack. The details of the transaction are saved until either the transaction session is committed or abandoned or the transaction is cancelled.

When a payment item (also known as a settlement product) is transacted successfully, Sell Product invokes Settle Transactions to determine whether the current transaction session can be settled. A payment may have been invoked directly from Settle Transactions for automatic payments or indirectly from the Settlement menu for manual payments.

FUJ00171948

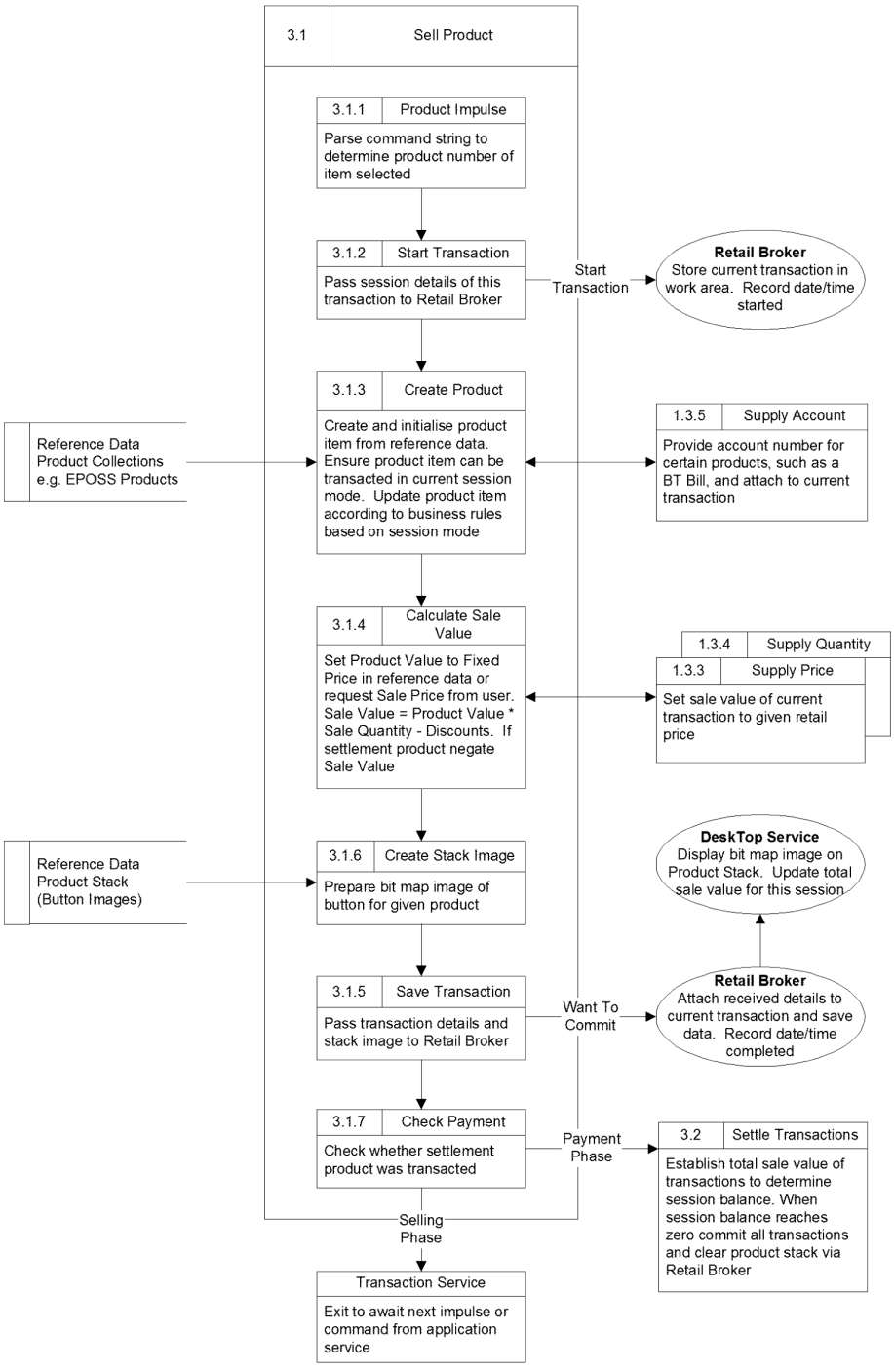


Figure 66 –Computational Model for Sell Product

FUJ00171948

5.3.5.5 Settle Transactions

The computational model for the Settle Transactions component of Transaction Session is given in Figure 67. When a payment item (also known as a settlement product) is transacted successfully, Sell Product invokes Settle Transactions to determine whether the current transaction session can be settled. Settle Transactions is also invoked when the user selects the Finish control on a product menu. See 5.3.3.10.3.

A manual settlement involves one or more payments managed by the clerk when a customer transaction is being processed. Customer transactions are serviced in Serve Customer (SC), Bulk Input (REC) or New Reversal (RV) modes. In these modes the Settlement menu is displayed to enforce manual payments. When the user makes the first payment, the payment phase is initiated. This is summarised by Settle Payment in Figure 67 and decomposed further in section 5.3.5.6.

Automatic settlement is managed by Settle Transactions for modes involving office transactions and remittance or transfer of items to and from stock units created at the outlet. In all non-customer modes a settlement product exists that enables unpaid transactions to be settled immediately.

Settle Transactions invokes Sell Product when the current session mode is associated with a known settlement product. In this case a settlement product is transacted with a sale value equal to the outstanding balance. Otherwise if the selling phase is still in progress the method of payments menu is displayed to enable the user to make manual payments, as described above.

When an outstanding balance of zero is determined after automatic payment during a ROAD session, Settle Transactions passes control to the LFS application. A generic design has been devised for the implementation of this solution so that any application can be invoked during the settlement of a session in a given session mode. The implementation is driven by an interface defined by Mode Parameters in Reference Data, giving the command string to pass to an application service. The return interface is supplied by Settle Transactions.

The current session is registered by the LFS application in the message store so that cash flows can be controlled at each outlet. A response code is returned by the LFS application indicating success or failure. If successful, Settle Transactions continues with the settlement of the transaction session. If the return code indicates failure the user is prompted to either abandon the session or retry settling the session via the LFS application. . The interface between Transaction Session and LFS is specified in EP/IFS/001.

A transaction session is committed to message store, via SettleAdjustedTxns in Retail Broker, when the outstanding balance of underlying transactions is zero. When committed the transaction session is complete and the Product Stack cleared, by Retail Broker, ready for the next transaction session.

After committal Settle Transactions checks whether a session receipt should be printed. The rules for generating a session receipt are governed by Reference Data except when an existing reversal is in progress for an APS transaction. In this case a session receipt is NOT produced.

Otherwise a session receipt is printed if the current session mode requires one as defined in Product Modes of Reference Data. A mandatory receipt is also produced when certain products are transacted in Serve Customer and New Reversal modes.

FUJ00171948

This is regulated by the Session Receipt attribute in EPOSS Products of Reference Data.

FUJ00171948

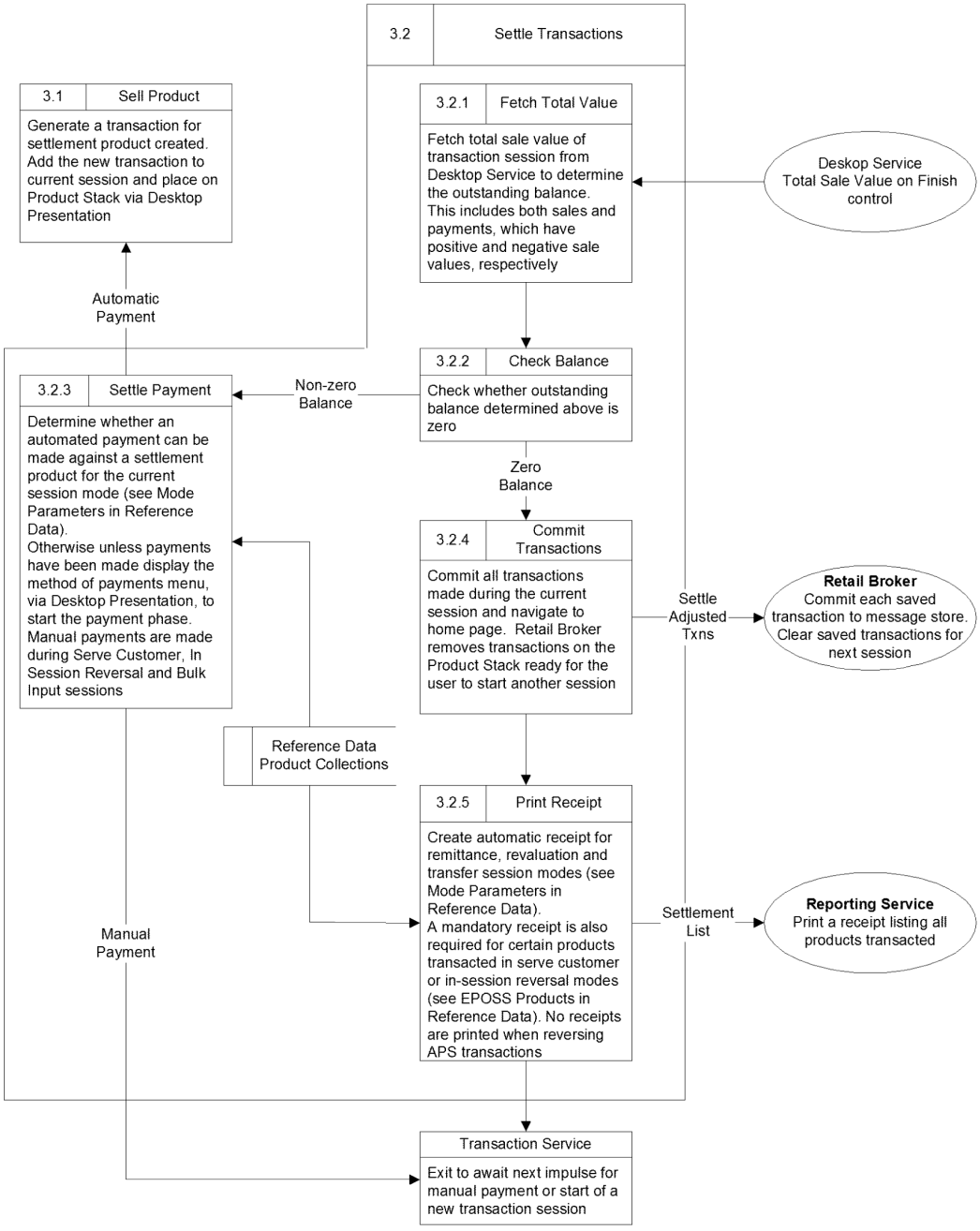


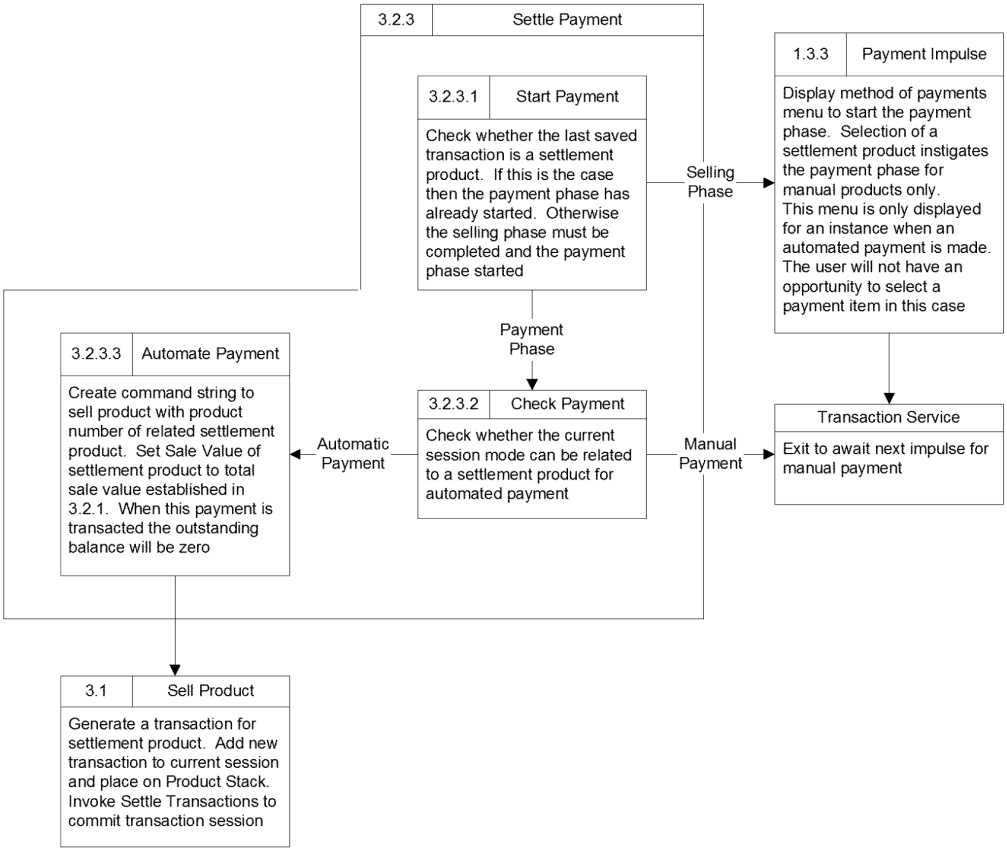
Figure 67 – Computational Model for Settle Transactions

FUJ00171948

5.3.5.6 Settle Payment

The computational model for the Settle Payment component of Settle Transactions is given in Figure 68. The selling and payment phases are clarified with respect to manual and automated payments already described.

Figure 68 – Computational Model for Settle Payment



FUJ00171948

5.3.5.7 Transaction Data

5.3.5.7.1 Business Components

The business rules for generating a transaction are implemented within Create Product and Calculate Sale Value components of Sell Product described and illustrated in section 5.3.5.4. These elements require further decomposition so the underlying business component of each element is devolved to form a central unit called Transaction Data illustrated in Figure 69. The figure shows Transaction Data as a central part of both Create Product and Calculate Sale Value components.

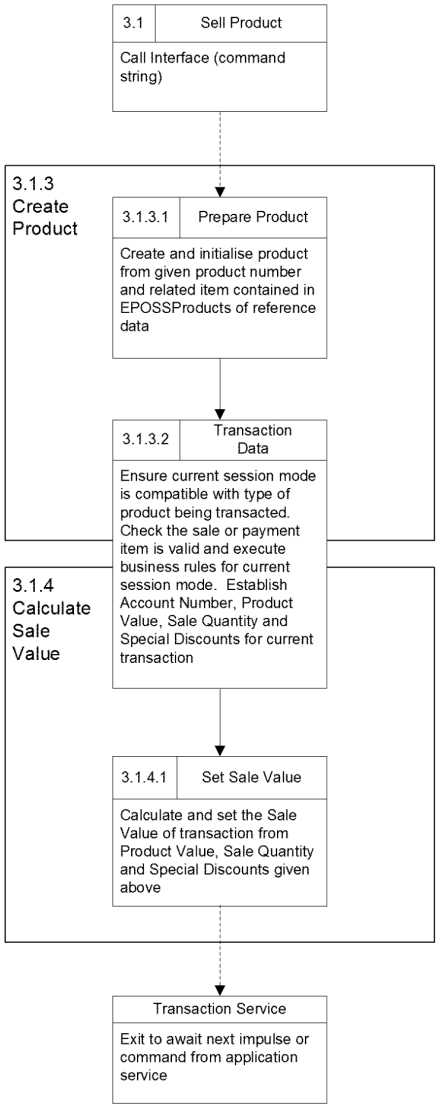


Figure 69 – Business Component of Transaction Data

FUJ00171948

5.3.5.7.2 Business Logic

The business rules for generating a transaction are implemented within Transaction Data, described in section 5.3.5.7.1. The logical components of Transaction Data, from the business perspective, are illustrated in Figure 70. Check Transaction, Evaluate Transaction and Process Transaction form the business logic of Transaction Data.

Generic business rules that are executed for all types of product are pre-processed and post-processed in Check Transaction and Evaluate Transaction, respectively. Specific business rules associated with the current session mode are handled in Process Transaction. Process Transaction is further subdivided into the types of session mode.

Transaction Data is driven by Reference Data collections held in the message store. For each transaction, product collections, in Reference Data, are extracted by Prepare Product to establish the attributes and properties of the product or service involved. Product collections are analysed by Prepare Product to execute business rules for a given product or service during a transaction session. For example preconditions may prevent a product from being transacted or additional data may be required.

Transaction Data also has to take account of the session mode during the transaction of a sale or payment item. The validity of a transaction and its results is dependent on the session mode and the associated attributes and properties of the product or service involved contained in product collections defined by Reference Data.

Product collections, held as Reference Data, identify the business objects and define the business rules for their transaction. Transaction Data utilises the given business objects and interprets the business rules to ensure they are implemented for each transaction generated by the system. See section 4 for an overview of the business rules imposed by Reference Data.

The logical components of Transaction Data are placeholders for the set of business rules identified in section 4. A table of business rules and their execution within logical components should be provided in the next version of this document to guarantee compliance with requirements.

FUJ00171948

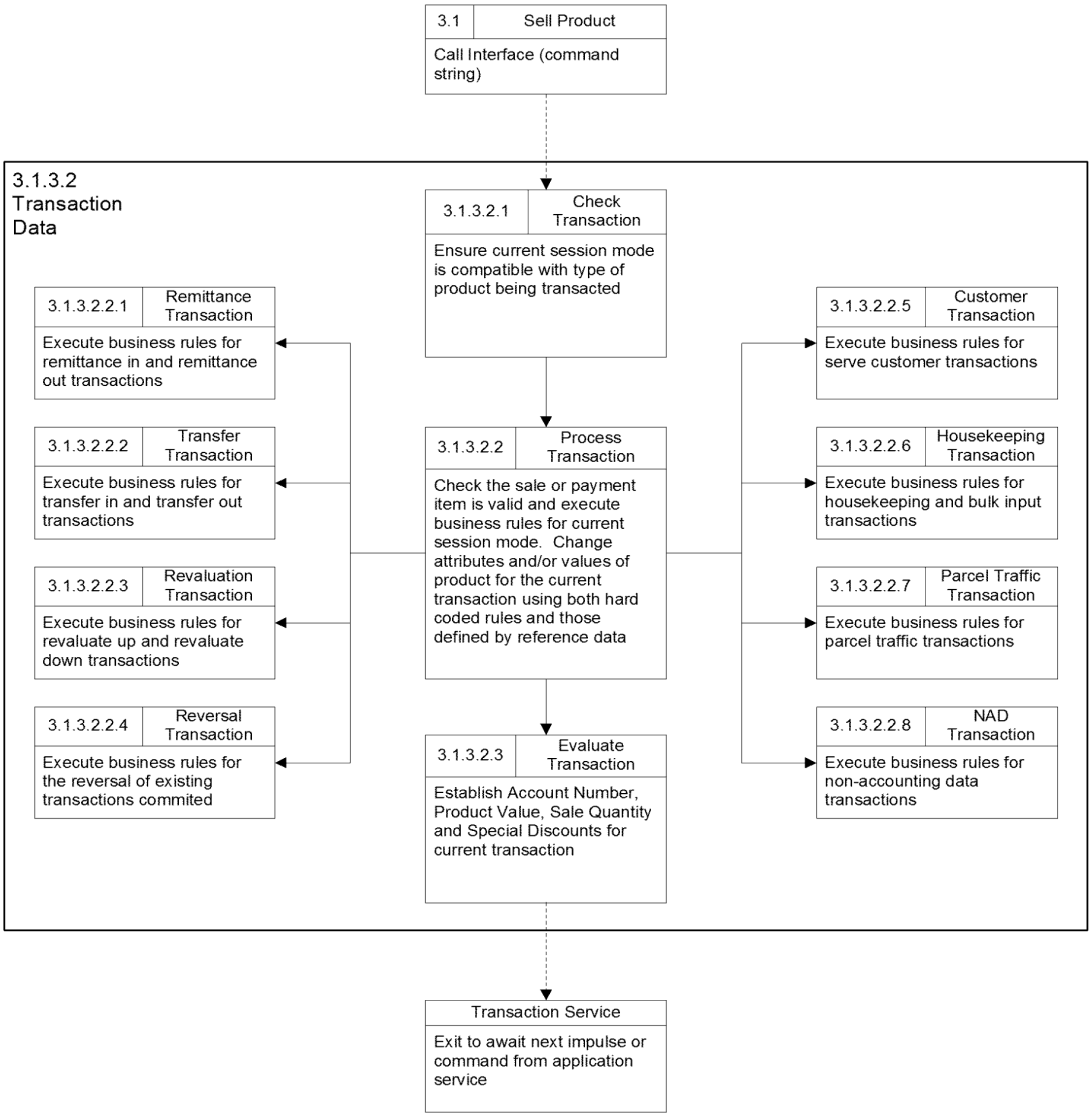


Figure 70 – Business Logic for Transaction Data

FUJ00171948

5.3.6 User Session

5.3.6.1 Swap Session

5.3.6.1.1 External to Internal Mapping

The clerk can suspend the current user session to initiate another user session and then swap between the original and new user sessions to perform more than one task at a time. No more than two sessions can be open simultaneously.

The current state of a user session can also be swapped from one counter to another when the clerk logs on at a different counter position as described in 4.1. A swapped session will continue in the normal manner from its current state at the new counter.

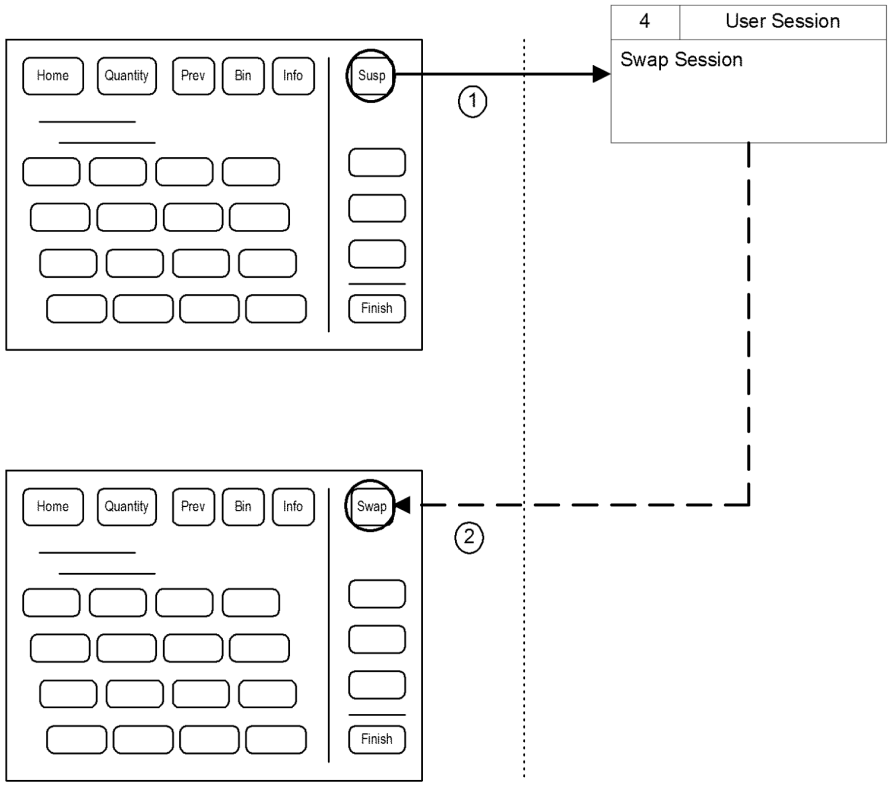


Figure 71 – Eternal to Internal Mapping of Swap Session

FUJ00171948

6 Networking

Transaction Service needs to know the state of connections between neighbouring nodes before certain operations are allowed to proceed. For example, a transfer in operation must ensure that all other users are locked out while the transfer out session is being utilised. The status of other nodes is checked beforehand.

The Riposte system is responsible for committing transactions into message store and ensuring data is replicated to correspondence servers. Transaction Service is not responsible for recovery in the event of a network failure.

FUJ00171948

7 Platforms

The EPOSS Transaction Service consists of two EPOSS component DLLs, namely Core and Settlement, developed in VB version 5. Transaction Service operates as part of the EPOSS Application Product in conjunction with a particular Riposte release and runs within the Windows NT platform using Internet Explorer. Therefore, there is dependency on the Riposte builds, Windows NT and Internet Explorer versions and service pack releases. The data relating to the versions and service pack releases may be found in the Physical Design for Counter at CSR+.

FUJ00171948

8 Systems Management

There is no involvement of the Transaction Service with Systems Management.

FUJ00171948

9 Application Development

This high level document defines the overall structure within which the developments of the Transaction Service have taken place. The major activity is in the evolution of the product, which implies keeping it up to date so that it continues to meet the business needs of ICL Pathway. It involves including new functionality.

The role of Application Development is to provide the EPOSS Application Product to support ICL Pathway's business strategy. As the EPOSS Transaction Service is an existing application, which has been developed, the role of application development is in the implementation of requirements introduced by Change Proposals and in providing support for problems arising from software. Product Development is therefore, carried out by code development and testing of the various EPOSS components.

Each Post Office outlet has different requirements, such as printing receipts in English or Welsh, or selling different lines of products. For this reason POCL requires the system to be easily customisable at each outlet. This is achieved not by changing the EPOSS code set, but by changing the Reference Data downloaded to the individual outlets.

FUJ00171948

10 System Qualities

The environment and architecture of Transaction Service has several inherent system qualities described in this section. Security, availability and performance are reliant of the underlying Riposte software. Potential for change and usability are the remaining system qualities considered.

Transaction Service provides an intuitive and easy to use set of menus and panels as part of user interface of the EPOSS application. Menu buttons are configured by Reference Data allowing products and services to be easily changed without affecting code.

Except for Reporting Service, there are no dependencies on other Counter Applications such as APS, OBCS, LFS and Balancing Service. The Reporting Service is invoked to print a mandatory receipt during the settlement of a transaction session. A receipt may also be printed manually by the clerk after settlement of a customer session.

Transaction Service provides a common interface for all applications to transact their products and services. Changes can be made to the underlying infrastructure without unduly affecting external application software.

All transactions must pass through Transaction Service, which acts as a single and central processing system for transaction management. This avoids duplication of effort in other parts of the system and prevents inconsistencies arising from different parts of the system.

Transaction Service is driven by Reference Data formulated by ICL and POCL.. This separates the business logic from the business rules and allows ICL to control EPOSS applications via configuration settings as opposed to code changes.

11 Solution Implementation Strategy

The EPOSS Transaction Service consists of two applications, which already exist as component DLLs, namely Core and Settlement. As these are already implemented in code, there is no solution implementation strategy. However, there is a strategy for the maintenance of the code, in that all updates to the code should be reflected in the appropriate design documentation.

FUJ00171948

12 Costs, Risks & Timescales

There are no costs and time scales involved in the implementation, as the Transaction Service is an existing functionality. There are risks involved if Pathway Change Proposals require alterations to the existing functionality. These risks will be critically assessed to minimise any negative impact on other applications. Change Proposals, which may impact the Transaction Service, will be thoroughly assessed.

The individual component DLLs, which constitute the Transaction Service, are designed to be used solely within the EPOSS Environment. There are risks involved in that many of the member variables and functions in the individual DLLs are exposed as public functions. Any applications with a handle to the individual component DLLs may access the public functions and methods. Variables that are exposed as public, and inadvertently set by client applications to erroneous values, could produce adverse effects on the correct operation of the Transaction Service. Interactions between the individual applications should be implemented via standard interfaces. The API designs will capture instances where these interfaces are not standard.

FUJ00171948

13 Appendix

13.1 Product Collections

13.1.1 EPOSS Products

Level	Attribute	Description	Format	Values
1	Group	FAD Code	Str(6)	Six numbers containing leading zeros used to reference organisational unit
1	Collection	Name for this collection of objects		"EPOSSProducts"
1	ObjectName	The identity of the object instance for each product item that can be transacted	Str(20)	See product number below
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	Rdata	Convention used for start of data block containing reference data	Null	Null
2	Data	Convention used for start of data block containing a set of records (defining each product item that can be transacted)	Null	Null
3	PN	Product Number giving primary system-unique identifier for the item	Str(10)	Primary Key

3	SN	A Short Name for the item	Str(10)	
3	LN	A Long Name for the item	Str(24)	
3	RN	A Receipt Name / Medium Name for the item – product name used when printing receipts	Str(16)	
3	MnV	Minimum Value of retail price that can be assigned to an item	Str(12)	Optional – not used for Items which are Logistics Inventory only
3	MxV	Maximum Value of retail price that can be assigned to an item	Str(12)	Optional – not used for Items which are Logistics Inventory (LINVI) only
3	MnQ	Minimum Quantity of an open priced item that can be transacted during a customer session.	Str(12)	Optional – always greater than or equal to one
3	MxQ	Maximum Quantity of a non-fixed price item that can be transacted during a customer session	Str(12)	Optional – always greater than or equal to one
3	LINVI	Logistics Inventory Item indicates that the item is an inventory item within the Logistics System	Str(5)	Optional. “TRUE” or “FALSE”
3	LACCI	Logistics Accounting Item indicates that the Accounting Item is for logistics purposes	Str(5)	Optional. “TRUE” or “FALSE”
3	ACCI	Product identity referencing the actual product used for accounting purposes	Str(5)	Reference to another product in this table
3	FP	Fixed Price indicates whether an operator can override the item price	Str(5)	“TRUE” or “FALSE”
3	IA	Null	Null	Not used
3	RF	Refundable indicates whether a refund is allowed against the item. Not used by Pathway	Str(5)	“TRUE” or “FALSE”

3	RP	Retail Price is the price of a single item including VAT, if appropriate	Str(12)	Optional – not used for items that are Logistics Inventory (LINVI) only. If FP is FALSE then this is the suggested price. Otherwise this is the actual price that must be transacted
3	SDI	Staff Discount Indicator indicates whether POCL staff are allowed discounts against item	Str(5)	Optional. "TRUE" or "FALSE"
3	CDI	Customer Discount Indicator indicates whether customers are allowed discounts against item	Str(5)	Optional. "TRUE" or "FALSE"
3	MV	The value that multiple sales of the item must be divisible by, e.g. Sales of National Lottery Instants must be divisible by £1. In other words this is the number of units of an item that comprise a single transaction	Str(12)	Optional – not used for Items which are Logistics Inventory (LINVI) only
3	AS	Adopt Settlement Sense indicates whether an item can change sign, e.g. a cash item or a cash equivalent item can be transacted either way	Str(5)	"TRUE" or "FALSE"
3	SE	Session Effect identifies whether an item increases or decreases an outlet's balance when transacted	Str(3)	"In" or "Out" where "In" is positive and increases an outlet's balance and "Out" is negative and decreases an outlet's balance (i.e. "In" => +, "Out" => -)
3	SR	Session Receipt indicates whether a receipt is mandatory when the item is transacted during a customer session	Str(5)	"TRUE" or "FALSE"- Always "FALSE" if item is an APS product or the number of receipts to be produced when this item is transacted is zero otherwise "TRUE"
3	Rcptno	Receipt Number used for identifying item on a printed receipt when a receipt is mandatory as defined by SR	Str(4)	Maybe zero

FUJ00171948

3	MP	Mandatory Product is the product identity of another product that must be included with the item being transacted – normally indicates one product only, i.e. no other product required	Num(10)	Reference to another product in this table. Null if no such relationship exists
3	V	Version number of the occurrence of this reference data entity	Num(10)	
3	SI	Service Instructions provide a message issued to the operator during a customer transaction	Str(30)	
3	VO	Voidable indicates whether this transaction can be voided during a customer session – typically only smart products will be non-voidable	Str(5)	"TRUE" or "FALSE"
3	I	Inventory item indicating whether this a non-zero value stock item	Str(5)	"TRUE" or "FALSE"
3	PM	Convention used for data structure defining Primary Mappings of Cash Accounts	Null	Null
4	L1	Accounting Node Mapping Level 1	Num(10)	Mandatory but may be Null
4	L2	Accounting Node Mapping Level 2	Num(10)	Mandatory but may be Null
4	L3	Accounting Node Mapping Level 3	Num(10)	Mandatory but may be Null
4	L4	Accounting Node Mapping Level 4	Num(10)	Mandatory but may be Null
4	L5	Accounting Node Mapping Level 5	Num(10)	Mandatory but may be Null
3	RA	Reversal Authority identifying the application responsible for authorising a transaction reversal		Optional. "APS" for APS products otherwise Null
3	RT	Receipt Type identifies a receipt definition for this item		Optional. A receipt type is provided for APS products otherwise Null
3	ST	POCL defined Service Type code for any transactions for this item		Optional. "B" for BES Products otherwise Null

FUJ00171948

3	PG	Product Group defines the product groups, in the form of a product identifier, categorising the item for selection from the pick list. Repeats for each product group	Num(10)	Reference to another product in this table
3	AdditionalData	Convention used for data structure defining each additional piece of information that must be captured when processing the item	Null	Null
4	P	Prompt describing the information being captured such as a message for an account number	Str(30)	
4	F	Visual Basic Wildcard comparison string used to validate captured data defining the Format of the additional information		"*"
4	S	Describes the type of Script being captured – Values are Numeric, Alphanumeric, Date and Currency	Str(20)	"Alphanumeric"
4	Max	The Maximum permitted value for additional information - used when validating the item	Num(10.2)	
4	Min	The Minimum permitted value for additional information – used when validating the item	Num(10.2)	
4	N	Name for the additional information being captured	Str(30)	
4	C	Caption for displaying additional information		
4	O	Number determining order in which additional information is captured	Num(2)	
4	VM	Error Message Text to display whenever the additional information is invalid	Str(100)	

FUJ00171948

4	A	Action allowed when the additional information is processed		"Display"
4	SD	Convention used for data structure containing a more detailed description of the additional information	Null	Null
5	Option	Convention used for data structure containing one entry for each option within an options list	Null	Null
5	Text	Text to display a description of the menu option	Str(30)	Null but see below for APS
5	Key	Key to display the value of the menu option	Num(2)	Null but see below for APS
3	PreCondition	Convention used for data structure containing zero or more pre-conditions for the item	Null	Null
4	PCProdNo	Product Identity of another item that must be transacted before this item can be transacted	Num(10)	Reference to another product in this table
4	ProductNo	Product Identity of item being transacted (see PN)	Num(10)	Reference to this product in this table
4	Msg	A message to be displayed if pre-condition is met	Str(90)	

FUJ00171948

13.1.2 PLU Impulses

Level	Attribute	Description	Format	Value
1	Collection	Name for this collection of objects		"PLUImpulses"
1	ObjectName	The identity of the object instance for each product item that can be selected from pick list	Str(20)	
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	RData	Convention used for start of data block containing reference data	Null	Null
2	Data	Convention used for start of data block containing a set of records (defining the pick list)	Null	Null
3	PLU	Product Lookup Number defining the product identity for the item	Str(30)	
3	PLUImpulse	Convention used for data structure containing each product in the pick list	Null	Null
4	PLU	PLU Number defining the product identity for the item		
4	Desc	A Long Name for the item		
4	SData	Convention used for data structure defining the button displayed on the Transaction Stack	Null	Null

FUJ00171948

5	StackCaption	A Short Name for the item		
5	StackPicFile	Path on the local machine for locating the bit map file displayed on the Transaction Stack	Str(30)	
5	StackPic	Name of the bit map file displayed on the Transaction Stack	Str(30)	
4	IntDef			
5	Cmd			“SellProduct”
5	ProductNo	Product identity of item being transacted		Foreign Key to EPOSS Products
3	Mode	List of valid modes for transacting the item. Repeats for each product mode		

13.1.3 Product Modes

Level	Attribute	Description	Format	Value
1	Collection	Name for this collection of objects		"ProductModes"
1	ObjectName	The identity of the object instance for each product mode	Str(20)	
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	Rdata	Convention used for start of data block containing reference data	Null	Null
2	Data	Convention used for start of data block containing a set of records (defining the set of product modes)	Null	Null
3	Mode	Convention used for start of data block containing a list of valid product modes. Repeated for each mode	Null	Null
4	M	A unique code to identify a type of Transaction Mode	Str(4)	"SC", "RIAD", "ROAD", …
4	V	The Version Number for the occurrence of the reference data entity	Str(4)	

FUJ00171948

13.1.4 Mode Parameters

Level	Attribute	Description	Format	Value
1	Collection	Name for this collection of objects		"ModeParameters"
1	ObjectName	The identity of the object instance for each session mode	Str(20)	
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	Rdata	Convention used for start of data block containing reference data	Null	Null
2	Data	Convention used for start of data block containing a set of records (defining the mode parameters)	Null	Null
3	ModeInfo	Convention used for data structure containing definition of mode parameters	Null	Null
4	IItem	Identifies the button on the menu initiating the change of session mode	Str(10)	
4	Cmd			"ChangeMode"
4	MaxStackTotal	Maximum amount of money that can be transacted during transaction session	Num(7.2)	"9999999.99"
4	Mode	Session mode used to initiate a transaction session		e.g. "SC", "ROAD", "RIAD"

FUJ00171948

4	MC	Indicates whether each product must be validated in this mode	Str(5)	“TRUE” or “FALSE”
4	SessionReceipt	Identity of Session Receipt	Num(5)	
4	SetlementProduct	Identity of settlement product when automated payments are made during settlement		
4	AlwaysPrintReceipt	Indicates whether a printing a receipt is mandatory after each session is completed	Str(5)	“TRUE” or “FALSE”
4	ReceiptTitle	Title on all printed receipts after settlement of transaction session	Str(30)	
4	CallApp	Convention used for data structure defining application interfaces	Null	Interface definition of another application called to process the change mode request (used by ROAD to invoke LFS)
5	InterfaceName	Convention used for data structure defining application calls	Null	Name of application and command string passed to standard interface
4	ReceiptHotKey	Indicates whether a hot key for printing receipts is available	Str(5)	“TRUE” or “FALSE”
4	ModeTitle	Full name of session mode	Str(30)	e.g. “Non Accounting Data”
4	ReverseSense	Indicates whether the sale value of each item transacted should be negated	Str(5)	“TRUE” or “FALSE”
4	DASS	Indicates the Accounting Settlement Sense	Str(5)	“TRUE” or “FALSE”
4	PrimaryMappings	Convention used for data structure defining Primary Mappings of Cash Accounts	Null	Null
5	L1	Accounting Node Mapping Level 1	Num(10)	Mandatory but may be Null
5	L2	Accounting Node Mapping Level 2	Num(10)	Mandatory but may be Null
5	L3	Accounting Node Mapping Level 3	Num(10)	Mandatory but may be Null

5	L4	Accounting Node Mapping Level 4	Num(10)	Mandatory but may be Null
5	L5	Accounting Node Mapping Level 5	Num(10)	Mandatory but may be Null
4	SecondaryMappings	Convention used for data structure defining Secondary Mappings of Cash Accounts	Null	Null
5	L1	Accounting Node Mapping Level 1	Num(10)	Mandatory but may be Null
5	L2	Accounting Node Mapping Level 2	Num(10)	Mandatory but may be Null
5	L3	Accounting Node Mapping Level 3	Num(10)	Mandatory but may be Null
5	L4	Accounting Node Mapping Level 4	Num(10)	Mandatory but may be Null
5	L5	Accounting Node Mapping Level 5	Num(10)	Mandatory but may be Null
4	NavigateString	Convention used for data structure defining menu of products for initiating session	Null	
5	Menu	Identity of source menu from which session mode is selected	Str(10)	
5	Menu	Identity of target menu when the session mode is initiated	Str(10)	

FUJ00171948

13.1.5 Desktop Buttons

Level	Attribute	Description	Format	Value
1	Collection	Name for this collection of objects		"DesktopButtons"
1	ObjectName	The identity of the object instance for each desktop button	Str(20)	
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	RData	The formatted attribute grammar defining the attributes of a desktop button	Str(1500)	

FUJ00171948

13.1.6 Product Groups

Level	Attribute	Description	Format	Value
1	Collection	Name for this collection of objects		"ProductGroup"
1	ObjectName	The identity of the object instance for each product group	Str(20)	
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	Rdata	Convention used for start of data block containing reference data	Null	Null
2	Data	Convention used for start of data block containing a set of records (defining product groups for categorising product items)	Null	Null
3	PGName	The product group name giving the category of each product	Str(30)	

FUJ00171948

13.1.7 Product Stack

Level	Attribute	Description	Format	Value
1	Collection	Name for this collection of objects		"ProductStackAttribute"
1	ObjectName	The identity of the object instance for each product item	Str(20)	
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	RData	Convention used for start of data block containing reference data	Null	Null
2	Data	Convention used for start of data block containing a set of records (defining desktop buttons for display on product stack)	Null	Null
3	StackCaption	A Short Name for the item	Str(30)	
3	StackPic	Path on the local machine for locating the bit map file displayed on the Transaction Stack	Str(30)	
3	StackPicFile	Name of the bit map file displayed on the Transaction Stack	Str(30)	

13.1.8 Tilda EPOSS Products

Level	Attribute	Description	Format	Derivation
1	Group	Fad Code	Str(6)	Six numbers containing leading zeros used to reference organisational unit
1	Collection	Name for this collection of objects		"~EPOSSProducts"
1	ObjectName	The identity of the object instance for each non-core product item	Str(20)	
1	Suffix	The version of the object instance	Str(2)	[00:99]
1	StartDate	The date when this instance becomes effective	Date / Time	Primary Key
1	EndDate	The date when this instance ceases to be effective	Date / Time	Optional
1	RData	Convention used for start of data block containing reference data	Null	Null
2	Data	Convention used for start of data block containing a set of records (defining each non-core product item)	Null	Null
3	Depend	Identifies whether or not an item applies to the outlet and is used to control non-core product usage	Str(5)	"TRUE" or "FALSE"