Lecture Notes

Structured Systems Analysis

Lecture 4

The Generic Model for Structured Analysis

Written by Dr. Fred Grossman

Copyright © 1999 Fred Grossman  All Rights Reserved
A Requirements Architecture and Patterns

Structured systems analysis provides an architecture for specifying system requirements without introducing any design decisions. The basis for a requirements specification is that it must be satisfied, there are no options. How the requirements are satisfied is left for the design, and depends upon the packaging alternatives selected. Most modifications to systems are design modifications not requirements modifications.

A pattern is a reusable, effective solution to a problem. A requirements pattern encapsulates expertise about a requirements architecture. It provides a means to model the requirements of virtually any system. The model that I am presenting is a **Generic Model** of a structured systems analysis using data flow diagrams. It is a generic requirements solution expressed in a pattern for a DFD. The model is instantiated for each system being analyzed.

**Generic Model**

The following DFD's introduce the generic analysis model. The basic assumption is that all (most) systems are really generically identical. There are five principal organizational system functions (only three are represented in this version of the model). A system **processes a request** for something. This might be a product or a service in the broadest sense. In order to satisfy the request, there will need to be some sort of resource available. This might be a product or a service that might entail supplying physical resources, personnel, information, training, etc. The system must **manage** the availability of these **resources**. All systems require some sort of **accounting** function. Accounting in the generic sense does not mean accounts receivable and payable, although it might in some instantiations. The remaining two generic system functions (not included here) are the pre-request function, e.g., marketing, and the post-request-satisfaction function, e.g. sales analysis.

While I believe that the application of this generic model to requirements analysis is a powerful approach for the resolution of the analysis problem of constructing the correct system, by no means is it a silver bullet. The success in developing (software) systems depends upon the application of "common sense" methodologies and solutions. Patterns and solutions have to be used in a way that balances the various forces. The generic model is a pure analysis model; if followed faithfully, it prevents the premature introduction of design decisions.
**Top-level Diagram**

1. **Requestor**
   - Manage request
   - Available resources
   - Internal resource accountability

2. **Resource Provider**
   - Manage Resources
   - New Resource
   - External resource accountability

3. **Request to process**
   - Processable request
   - Available resources
   - Internal resource accountability

4. **Processable request**
   - Make a processable request
   - Satisfy the request
   - Processed request
   - Internal resource accountability

**Explode Manage Request**

1. **Requestor**
   - Request
   - Processed request
   - Available resources

2. **Resource Provider**
   - Resource order
   - Resource receiving
   - Provider invoice

3. **Request to process**
   - Request
   - Available resources

4. **Processable request**
   - Processed request
   - Internal resource accountability
Explode Account for Resources

Account for Resources

3.1 Account for Internal Resources

Internal resource accountability

Requestor payment

Statement

Requestor

3.2 Account for External Resources

External resource accountability

Provider invoice

Provider payment
Order Processing Model

As an example of how to use the generic model, we will consider an order processing system. In many cases, this can be used as a pseudo-generic model since it is almost generic and is easier to relate to.

When applying the generic model to a systems analysis, the first step is to identify the "request", the "requestor" and the "resource" being requested. In the order processing example this is simple. The Customer is the requestor and the Order is the request. The resource being requested is represented by an Inventory of items being sold to the Customer. The satisfaction of the request is represented by the Invoice which describes the shipment of the ordered items. Remember that the "physical" goods are not shown in a DFD, only their data representation.

The accounting function is perhaps the most difficult to instantiate. Accounting in general does not mean the traditional accounting functions associated with a business, i.e., receivables and payables. However, in the order processing system this is exactly what the accounting responsibilities are.
**Manage Order Explosion**

1. Order → Make Shippable Order → Shippable Order

2. Order to Ship → Ship Order → Invoice
   - Available Inventory → A-R Invoice

**Make Shippable Order Explosion**

1. Order → Verify Order
   - Valid Order

2. 1.1.1 → 1.1.2
   - Determine Credit Status → Shippable Order
Manage Inventory Explosion

Order demand
Inventory Availability

2.1
Purchase Item

Purchase Order
Open Purchase Order

2.2
Receive Item

New Inventory
A-P Receiving

Open Purchase Orders

Manage Inventory

Purchase Item Explosion

Order demand
Inventory Availability

2.1.1
Determine What to Order

Item to Order

2.1.2
Order Item

Purchase Order
Open Purchase Order
Manage Accounting

Manage Accounting

A-R Invoice → 3.1 Manage Accounts Receivable → Customer

Customer Payment → Statement

Customer

A-P Receiving → 3.2 Manage Accounts Payable → Supplier Invoice

Supplier Payment
The Rental Generic Model

A variation of the generic model is needed to best describe the requirements when the request involves a rental (or use of an item with a guaranteed return). This is actually included in the generic model, but at a lower level in the diagram set. Any request satisfaction may on exception not be satisfactory, and thus require a return. The key term is exception. Normally returns have much less importance than requests (orders) and thus should not appear at the same level as the request (order). Remember that on any diagram, all items should be of the same relative importance.

The following top-level DFD illustrates the rental generic model.
Applying the Generic Model

The following example illustrates how to apply the generic model in a structured systems analysis. The system being analyzed is a tax audit system. An IRS field office selects tax returns to be considered for audit. The system we are analyzing receives tax returns from the field office, determines auditability, audits the appropriate returns, and sends the audit result to the field office.

The first task is to determine the requestor and the request. The requestor is the IRS Field Office. The request is a Tax Return to audit. Next we must determine the resources that are required to satisfy the request. In this case the necessary resources are the Supporting Documents for the Tax Payer’s tax return. If the documents are not available, they must be requested from the Tax Payer. After the audit is completed the IRS Field Office is sent the results. The system must ensure that the audit disposition is completed properly. That is the accounting function. Not the collection of taxes and penalties due nor the refund of over payment, however.