VisualAge Requirements Tool
Technical Overview
vrttech

April, 1996

J. Carlos Goti
Cathy Owen

IBM* Software Solutions
555 Bailey Road
San Jose, CA 95141
JCG at STLVM6
CHOWEN at STLVM6

First Edition (April 1996)

Preface

The Presentation

This presentation provides an overview, a high level technical description of the features in VisualAge* Requirements Tool (VRT) and its value to customers. The script has been written in a style intended to be spoken, as opposed to read.

The experience gathered from past customer presentations is:

- Customers see the applicability of VRT to their shops very quickly, especially if they see a demo.
- Some have said that they are trying to get a hold of their set of business rules and are struggling with the problem. They see VRT as a good tool to do that.
- Some have said that they wish they had a tool like VRT in their last requirements project. It would have saved them a lot of time and allow them to produce a better product.
- The CFO of a company saw much value (and signed up for the beta program) in interacting with other functional managers in his company.
- Consultants love the function. It allows them to do work very quickly, with minimal effort and look good in front of their customers.

Trademarks

The following terms, denoted by an asterisk(*) on their first occurrence, are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries:

IBM	VisualAge
OS/2	VisualGen
TeamConnection	ProModeler
DataAtlas	

Product or company names, denoted by a double asterisk(**) on their first occurrence, are trademarks or registered trademarks of their respective holders.

Questions or Comments

For questions or comments regarding this paper, contact:

J. Carlos Goti
IBM Corporation
555 Bailey Avenue
San Jose, CA 95141
(408)463-2897
8-543-2897
JCG at STLVM27

Cathy Owen
IBM Corporation
555 Bailey Avenue
San Jose, CA 95141
(408)463-3069
8-543-3069
CHOWEN at STLVM6

Contents

VisualAge Requirements Tool - Technical Overview 1

VisualAge Requirements Tool Objectives 4

Benefits to Business People 5

Benefits to Applications Developers 6

Systems Evolution 8

Requirements Cycle 9

VisualAge Requirements Tool Elements 10

VisualAge Requirements Tool Overview 11

Cross Business Operations Relationships 12

Methodology 13

Specify Business Transactions 14

Specify Business Information 15

Specify Business Facts 16

Data Dependency Associations 17

Business Information Dependency Diagram View 18

Specify Business Rules 19

Summary of Associations 21

User Role Diagram 22

Business Transaction Flow Diagram 23

Consolidate Data 24

Review Your Work with Your Users 25

Team-based Applications Development 26

Scenario: Integration with VisualGen 27

VisualAge Requirements Tool to VisualGen Mappings 28

Process Tools and VisualAge Requirements Tool 29

Scenario: Usage with ProModeler 30

Import/Export Facilities 31

Standard and Professional Editions 32

Engages the Business Community 33

Engages the Development Community 34

Summary 35

VisualAge Requirements Tool - Technical Overview

VisualAge Requirements Tool is a tool to assist in the development of descriptions of system behavior. It can be used to specify automated systems (applications) or manual systems (departments). These behavioral descriptions are used to:

- specify a new system (typical development phase)
- specify an existing system (re-development and re-use projects)
- specify a system you wish to sell (advertise your function in a clear way)

These specifications can be used as the famous-but-never-seen *contract* between consumers and suppliers of application systems. They can be used in development and bid or buy situations.

The aim of VisualAge Requirements Tool is to create specifications of system behavior that business personnel can understand and validate. In Information Technology there are other means of specifying function, but they are all targeted to IT and not users, except for manually drawn diagrams and textual documents. There is no competition to VRT at this moment. The test to determine if a tool competes with VRT is very simple. Ask the question *Does this tool help business users develop and validate behavioral specifications for a system?* If the answer is no, then the tool does not compete with VRT.

VisualAge Requirements Tool gives meaning to the concept of business rules. The current discussion of business rules in the industry has centered around data rules and process rules. But these rules are expressed in IT terminology and they lack the context to make them meaningful to business personnel. By adding the concept of transaction into VRT′s model, the context to connect rules to business events or processes is provided.

VRT not only exposes rules to the business person in a meaningful context, but it has a built-in classification for business rules. This enhances both the discovery of rules and their subsequent management, once they are stored in the tool′s repository. Business rules are also attached to business objects, to further define their function and place them in an additional context for understanding and completeness validation.

VisualAge Requirements Tool is s tool whose niche is very significant. Many millions are being spent in developing systems that never get installed in production or require massive rewrites. The top-ten list of inhibitors to effective development projects includes *lack of user involvement*, *poor/incomplete requirements* and *lack of common vision about the project*. These areas are addressed directly by VRT and the early customer feedback is very positive as to its effectiveness.

Publications related to VisualAge Requirements Tool

- **Product Documentation**

  IBM VisualAge Requirements Tool for OS/2 fact sheet, GC26-8229-00

  IBM VisualAge Requirements Tool for OS/2 User′s Guide, SC26-8228-00

  The online tutorial for IBM VisualAge Requirements Tool. The tutorial is installed when the product is installed.

- **Fundamental ideas to justify the approach and help in the real-world use of VisualAge Requirements Tool are described in the following technical reports which are available through the IBM Library systems.**

  Business Rules: Their Use for Business Operational Analysis, by J. Carlos Goti, STL TR03.550, April 1994.

  Finding Business Objects in Legacy Applications, by J. Carlos Goti, STL TR03.561, May 1994.

A Business Classification for Business Rules, by J. Carlos Goti, STL TR03.563, June 1994. This paper was published in Database Newsletter Sep/Oct 1994.

Business Rules: Their Role in Systems Requirements, by J. Carlos Goti, STL TR03.587, October 1994.

- **The following IBM ITSO red-books are available from PUBORDER.**

  Business Process Reengineering and Beyond, IBM International Technical Support Organization (ITSO), San Jose, December 1995 - order number SG24-2590.

  Introduction to the Applications Development Team Suite. IBM International Technical Support Organization (ITSO), San Jose, February 1996 - order number SG24-4648

- **The following book on OO modeling proposes business transactions and business rules to extend use-cases.**

  Visual Modeling Technique: Object Technology Using Visual Programming by Daniel Tkach, Walter Fang, and Andrew So, Addison Wesley, May 1996, ISBN 0-8053-2574-3.

- **Books that address the area of requirements.**

  Software Requirements - Revision - Objects, Functions, and States by Alan M. Davis (University of Colorado), Prentice Hall, 1993, ISBN 0-13-805763-x.

  Exploring Requirements - Quality BEFORE Design by Donald C. Gause and Gerald M. Weinberg, Dorset House Publishing, 1989, ISBN 0-932633-13-7.

  Object-Oriented Software Engineering - a Use Case Driven Approach by Ivar Jacobson, Addison-Wesley, 1992, ISBN 0-201-54435-0.

  Essential Systems Analysis by Stephen M. McMenamin and John M. Palmer, Yourdon Press, 1984, ISBN 0-13-287905-0-025.

- **A reference to the state machine model, that can be considered as a foundation for our model.**

  Principles of Information Systems Analysis and Design by Mills, H.D., Linger, R.C. and Hevner, A.R., Academic Press, Inc., 1986, ISBN 0-12-497545-3.

- **Business rules publications.**

  The Business Rule Book, by Ronald Ross (Editor and Publisher of the *Database Newsletter*) was published in 1993.

  The periodical *Database Programming and Design* featured articles over the past two years by Terry Moriarty and Barbara Von Halle.

GUIDE has a Business Rules project, which presented its report in November 1995. This project concentrated on the expression of rules at the IT level. A new project was started to develop the definitions and properties of business rules, at the business user level. J. Carlos Goti is the IBM representative on both projects.

The following types of customers were exposed to various levels of the technology and tool:

- Banks
- Insurance companies
- Oil companies
- Airlines
- Manufacturing
- Etc.

External presentations:

- GUIDE - presentations in July and November 1994 and March, July and November 1995.
- 4th RE-ENGINEERING FORUM, Victoria, BC, in September 1994.
- CASCOM, Toronto - IBM executives and customers, in October 1994.
- "The 1996 Business Rules Summit", San Francisco in February 1996.

VisualAge Requirements Tool Objectives

The state of the technology and industry can be summarized with:

1. Business Rules are the technology with the highest benefit potential (T. Percy, in Systems Software Architecture, Technology, December 27, 1995).
2. Helping business users participate in clear requirements is of high value to development shops, which have high development failure rates (Jim Johnson, Application Development Trends, January 1995).

The objectives of VRT can be summarized in the following value statements.

- Avoid development mistakes at the earliest possible point.
- Enable the management of business rules to ensure re-use and a clear evolution with business changes.
- Enable to identification of program elements that implement each requirement to facilitate application updates and other controls.
- Implement a solid analysis method to develop better requirements.
- Create a common understanding between users and the IT organization.

Benefits to Business People

- Assist business users to determine/discover what they want the system to do in relation to the operations of their business.

  Users of systems can be very critical of systems that support them but can′t specify all needed functions before they have the system to exercise. This causes the development of many systems that do not measure up to expectations when delivered. Users have a hard time specifying systems ″ahead of time″ because the insertion of a new system in their business operations alters those business operations, in ways that are hard to anticipate, generating additional requirements and making some of the system′s functions obsolete.

  One of the ways to address this problem is through prototyping (upper-CASE tools have proven not to solve the problem because they are too systems oriented in their information and presentations), which requires the mock-up implementation of the application to show users some of the most important system features. Users can then exercise the prototype and ″simulate and evaluate in their minds″ how the system would be used if it were real. Some prototypes emphasize only the user interface to evaluate a proposed look-and feel. ″Low fidelity prototypes″ are used for this purpose.

  VRT offers an alternative that does not emphasize user interfaces but functionality (internal system behavior). It identifies the system′s services through the concept of transaction and the behavior of those services through the concepts of business rules and business information object. Once the requirements are specified, business users can simulate (in their minds) the behavior of each of the specified services and evaluate them from a business functional viewpoint.

  The representation of a system′s behavior using VRT should take a fraction of the time that it would take to prototype the same information. VRT also provides various facilities to check the completeness and consistency of the requirements, thus enhancing the confidence level of the specification. Changes can easily be made and validated for quick turn around in this dynamic activity. VRT also offers easy-to-enter annotation facilities to allow the collection of reviewers′ comments.

- Assist business users to specify the behavior (requirements) of the system they need.

  The same information that users need to ″discover″ the requirements of a system can be used to record those requirements. That is, there is no time lost in restructuring the information needed to understand and discover into what becomes the true specification of behavior.

  The completeness support provided by VRT gives high confidence that the specifications represent what is really needed.

- Provide business users a semi-formal representation of system behavior to communicate among themselves and with systems providers.

  Once business users understand their requirements they need to express them in a consistent manner for use by information technology analysts. Traditionally, in the applications development process this step is called ″analysis″.

  In relation to application systems, business users work in what some people call the ″problem domain″ and systems personnel work in the ″solution domain″. These domains have their own terminologies and conventions. VRT is designed to work at the boundaries of these domains and has easy-to-understand basic terminology. VRT users define business elements by introducing the terminology of their choice to describe operational and behavioral aspects of their requirements specification.

  The same VRT information that was used to understand the business needs serves to express the results of the ″analysis″ phase of applications development.

Benefits to Applications Developers

- Provide systems developers with a behavioral requirements specification that can be used as a source of systems analysis information.

  The information contained in a VRT representation is used to specify analysis results to various application development needs:

  VRT provides the following elements to data design:

  - Identification of entities (business information) and their attributes (business facts)
  - Identification of relationships between entities (dependency associations)
  - Constraints on all of the above (defines rules)

  VRT provides the following elements to procedural program design:

  - Identification of all required program functions (groups of transactions that are connected through business rules)
  - Specification of the program functions (dynamic rules and their data retrieve and update requirements)
  - Program function re-use (support for rule uses rule)

  VRT provides the following elements to object-oriented design:

  - Identification of use-cases and user roles (transactions and their user role presentations)
  - Identification of business object classes (business information), their states (attributes and defines rules), their associations(dependency relationships), and methods (dynamic rules).
  - Identification of function sequencing classes (transactions and the sequence of rule they trigger)

- Provide systems developers with a behavioral requirements specification that can be used to develop functional verification test plans.

  Functional verification is the activity through which system providers show system users (business personnel) that the functions specified in the requirements statement are satisfied by the system under evaluation. The VRT representation provides a clear way to do this.

  Every transaction should have test cases that show every possible behavior as determined by the specified rules. Behaviors are very simple to express because the lasting results of an input transaction are the outputs transactions that it produces (under rule control) and the effects on the system's persistent memory (also under rule control).

  The basic plan is: for every input transaction create a set of test cases that exercise every condition dictated by the rules involved. Ensure that the appropriate transaction(s) are produced and that the persistent memory (business information) reflect the appropriate changes.

- Provide systems developers with a behavioral requirements specification that can be used to calculate function points and thus develop an early cost estimate for the new system.

  Function points are widely used in applications development to do two things: develop cost estimates for system development and measure development productivity. The elements in the VRT model provide enough information for the calculation of function points for the purpose of doing an early cost estimate.

  Function points are a single number that expresses the functional contents of a system. They are calculated by identifying all input and outputs (transactions), data accesses (rules that use business information), and files or data bases (business information). Each of these elements is given a complexity factor by an analyst and these number are added up by category. The totals for each category are multiplied by a weight and all numbers are added up one more time.

  The inventory provided by VRT representations has all the raw material to support the calculation of function points.

- Provide a basis for system′s evolutionary changes involving both business and systems personnel.

  A system developed using VRT for the expression of requirements is well positioned for future changes and evolution. New functions are represented as new transaction sets. The behavior of those transactions may be already supported by existing business rules. Or at least, one can validate new behaviors against the ones that already exist.

  Changes in existing transactions can be pinned down by analyzing the business rules affected and impact analysis is carried out at the ″business level″ to evaluate changes that ripple through the system′s functionality.

  Actually, these activities coupled with the capability of calculating function points provides a very strong basis for making decisions relating function and cost. One can do what-if evaluations using the information contained in VRT.

- Act as a business oriented index to implementation components for the location and reuse of business rules and other elements.

  VRT representations do not display the design of the system. They are expressions of requirements (in the business domain). For example, transactions identify the need for an interchange between the user and the system but this interchange could happen in many different ways: a business form, an NPT screen, a window/mouse based interaction, etc.

  But, the requirement still need to drive the function provided by the system. There is a correspondence between requirement based elements and implementation artifacts. Examples are:

  – Transactions: GUI′s, NPT screens, CICS or IMS transactions, printed reports, etc,
  – Business rules: program modules or procedures, data constraints, etc.
  – Business information: data bases, files, etc.

  A system that keeps track of these correspondences provides significant functions that make re-use and impact analysis a reality from the external/business point of view. One can tell where business rules are implemented and quickly change them to support the evolution of the business itself.

  Extra development efforts to make some of the parts re-usable are well spent because parts are visible and easy to find. They do not disappear in libraries never to be seen again because no one knows their role in the system, from an end user point of view.

Systems Evolution

This is a known cycle showing the evolution of systems (business automation). The main message here is that the evolution cycle is repeated many times over, even in a single business, it has a strong impact on the business, and we should attempt to perfect it as much as possible. We want to engage in this cycle of activities because there is a business change and not because a system got developed poorly and needs to be modified to make it usable, wasting resources and creating untold friction between users and IT.

VRT engages the user in the development project and helps assure a much better specification of the work to be done. When a wave of changes comes for a system that used VRT on a prior wave, the situation is very favorable. The user can review existing function, determine exactly where changes are needed, and have IT understand the needs in precise terms. IT can see the impact of the changes, from the requirements side and determine what implementation elements are affected

Change control is provided as a base function of VRT.

Requirements Cycle

The specification activity in the previous diagram is analyzed here, keeping in mind the facilities offered by VisualAge Requirements Tool.

The suggested sequence (with some iterations) is:

1. The analysis is done through interactions with the set of users involved in the specification. This phase is one of recording objects as they are discovered but we suggest the following *preferred* sequence:
   - Identify transactions and their data contents
   - Derive the information dependency diagram using the data content of transactions and traditional functional dependency grouping of data elements.
   - Identify the static rules for transactions and information objects.
   - Identify the dynamic rules using information and transactions as a guide. Utilize the built-in rule classification in the discovery process.
2. Consolidate the data using diagrams, and tracing facilities.
3. Submit the requirements specification to users for validation and corrections. If the users have access to VRT, the submission can be the transfer of file exported from the base tool to be imported by the reviewer′s tool.
4. The developer of requirements should consider holding JAD sessions with users to benefit from the synergism of the expert′s interactions. VRT can be used to display the specifications as they are discussed. Corrections can be made on the spot, if the task does not disrupt the proceedings.

VisualAge Requirements Tool Elements

In order to create specifications that are understood by business users, VRT has a very limited set of elements. They are named using common terms.

1. Business operation: is simply a way to divide and conquer the complexity of business systems. It allows users to address one aspect of the business at a time (for example accounts payable or order entry). They are recognizable area of the business function, either manual or automated.

   Business operations enable the focusing of concerns in a part of the business and should be named after the systems already existing in the enterprise. A business operation is considered a black box from the viewpoint of its implementation. The user only cares about its external manifestations (transactions) and the ways in which the system responds to them (rules).

2. Business transaction: is an interaction between the user and the business operation. Knowing all the transactions (input and output) that a system accepts or provides is a very powerful way of specifying its scope. A function is not provided if a corresponding set of transactions does not specify it. There can be no surprises when the system is delivered because the user has seen the set of transactions that were planned for it at the time the requirements were specified.

   Transactions are further specified by their connections to business facts (data elements for IT). A transaction can be thought of a business form exchanged between business entities that require support from one another. They are the manifestations of the mini-contracts between the client and supplier of a business function. In some cases the supplier of a business function is an application (automated function), in others the supplier is a person (manual function). At the level of definition used in VRT there is no distinction between these two.

3. Business information: is the representation of a business object of interest to the definition being specified. We all know that applications (and file cabinets) keep information about objects of interest to the enterprise. And we know that the information is kept to aid in the execution of business tasks. But current information has a price: it has to be created and updated as the real-world object that they represent changes over time. The company needs specifications explaining how these update functions are carried out. Business rules are the elements that do that.

4. Business rule: is an expression of a desired property or behavior of transactions and information elements.

   Static rules (defines association, in the next diagram) specify properties of transactions and information elements. They say things like *in a check (transaction) the written amount must match the numeric amount* and *and order (information) must have a customer (information) associated with it*.

   Dynamic rules allow the specification of behavior under different conditions. They are associated with transactions and can initiate transactions (produce output) and look at or modify information (carry out update functions).

   Business rules are expressed in plain text. They have a format, just to to normalize their appearance for ease of understanding. A given business operation may have lots of rules (hundreds or thousands) and VRT has a classification to help users manage such large numbers. Also, the different views provided allow the user to only look at the rules of interest to the task at hand.

5. Business fact: is a data element in IT terminology. Facts are needed in requirements, not only to specify the contents of transactions and information elements, but to introduce a standard terminology for the textual expression of rules. For example, a rule may need to say *indicate this order is late*. In order to do that we need the concept of order (information) and of some attribute of order that may be called the ″order late indicator″.

VisualAge Requirements Tool Overview

Using the elements described in the previous diagram, we can create business meaningful specifications connecting them through associations.

A business operation can contain transactions, information, rules and other business operations. Transactions, information and rules can not exist unless they are contained by some operation. The user is seen as sending or receiving transactions from a business operation, through the transactions contained in it.

Facts can be seen through a VRT dictionary facility, where they are all created and deleted from. Facts can be composite to facilitate grouping of business data concepts (the largest grouping of which are transactions an information). And facts can be included in transactions and information elements. Notice that facts are not contained in an operation; this was done to create an enterprise wide fact dictionary, one that does not change from operation to operation.

Information can have dependencies on other information elements. A dependency represents a business visible relationship between the business objects represented by the information elements. For example a check (information) has a dependency on account (information).

Static rules connect to transactions and information through the defines association.

Dynamic rules are triggered by transactions and in turn interrogate and/or modify information and initiate transactions.

Rules can use other rules to support rule re-use.

Cross Business Operations Relationships

The business operation element is a facility that encapsulates groups of business function. It contains transactions, information and rules and most of the associations that are defined between them. A few exceptions to this encapsulation paradigm were allowed to enable re-use.

An information element can have a dependency on an information element in a business operation different from the one it is contained in. This facility reflects the common occurrence of operations sharing the information they have with other operations.

A business rule can use (invoke) a rule in an operation different from the one it is contained in. This allows the re-use of the function encoded in a rule.

For reasons mentioned above, the Fact Dictionary does not belong to any business operation and its elements (facts) can be included by transactions and information elements in any business operation.

An operation external to another is seen as an external user by the first. That is, they can exchange transactions.

Methodology

This is pretty much self-explanatory, especially with the material covered before in these scripted notes.

Specify Business Transactions

Transactions are the only way that a business operations connects with the world of its users. And transactions are the anchors for the specification of any function planned for the operation. Transactions can be received or sent (by an operation) to a human user or to an automated user.

Remember what was said before: an operation will not perform any function that is not specified by a transaction. The correct and complete identification and specification of all transactions is what the requirements process and VRT are all about.

Specify Business Information

IT people have a strong tendency of looking at information elements as entities with their associated attributes (by including facts). This conception is correct if we think about the *implementation* of information elements. The intent in VRT is somewhat more abstract.

Each information element represents a business object that the operation needs to maintain. This is well known to business users. For example a manager keeps information about his/her employees, information about the company assets under his/her control and most people keep information about their bank accounts.

A Business operation should only keep information about the objects that are handled by it. These objects are externalized through transactions. We can say that an operation should only keep information about objects *visible* through some transaction that it supports, and no more. Additional information is not relevant and should not be included in an operation, it would only increase complexity and confuse the business user.

Specify Business Facts

Business facts included by an information element indicate what type of data is to be kept by that information about the corresponding business object.

Notice that, except for some special cases, all facts in the set of information elements of an operation should appear in some transaction issued or received by the operation. Why would the operation keep data that is never seen by its users?

Also, since we should attempt to have a *normalized* view of the business data, a fact that is included by an information element should not be included by others.

Data Dependency Associations

This foils summarizes the information presented in the previous ones except for the definition of the *depends* association. The depends association between two information elements indicates that there is a *business visible* connection between the objects involved.

Depends associations are also detected by analyzing transactions and the facts they include. If two objects appear together in a transactions, there is a strong indication that there is a dependency between them. The dependency is implied by the user of the transaction. For example the transaction called bank-check shows the customer name and the account number - this implies the need to have a dependency between customer (information) and account (information).

To keep the specification as simple as possible, no dependency should be added unless there is a transaction that requires it.

Business Information Dependency Diagram View

The Information Dependency diagram view is of fundamental importance to the process of discovering/defining business rules. As we said before, rules are about business objects and this diagram shows all business objects of interest. There can be no rules about anything else but what is shown here (except for define rules on transaction, but they are easy to find).

The user can look at all information elements and analyze their importance to the operation. Some are crucial (the operation updates facts in them), others are marginal (read-only and imported from other application systems) and require little attention.

Most business objects have specific experts, who can be scheduled for rule definition sessions after the analysis just mentioned.

Specify Business Rules

An important aspect of developing a set of rules is having a pattern to discover them. Another is to create rules that are re-usable.

Rule discovery

1. Static rules can be derived by addressing each of the transactions and the information elements in the operation. Rules about information can allude to the dependencies that start or end at that information element.

   Rules that define transactions can be about:

   - Their presentation format
   - Their required date and time or frequency.
   - Computations involved in the derivation of values
   - The user role that receives or sends it.
   - etc.

   Rules that define information can be about:

   - Valid values of their facts
   - Number of entries
   - Attributes of the associations
   - etc.

2. Dynamic rules are triggered by transactions and interrogate or modify data and initiate transactions. So, in theory, if we had all transactions we can identify the rules that specify each of their behaviors and be done with it. In reality, we want to have additional approaches to help us reach the feeling of having done a complete job.

   An additional way of discovering dynamic business rules is to resort to the information dependency diagram for a list of business object and their dependencies. We can think of rules that govern the life of each object, regardless of the transactions that trigger the behavior and discover, as a by-product, transaction we have omitted.

   The built-in classification for business rules gives us powerful assistance in the second discovery process. Given a business object, and with the object experts at hand, you can use the classification to focus the team′s thinking in a rule category at a time. Rules will be discovered very quickly.

Rule re-usability

For this section we can borrow ideas from OO design. Elements are re-usable when they have a simple function and can be encapsulated in a meaningful way.

Rules are most re-usable (through the rule-uses-rule association) when they relate to a single business object. They will tend to represent a single business function. In our *order entry* example, the create quotation rule *uses* the allocate inventory and the price quotation rules. Notice that allocating inventory and pricing a quotation are re-usable functions because there may other situations in which a price is needed (given items and their quantities) and the allocation of inventory to a demand required.

Some rules would not be complete if they only relate to a single business object. In these cases we have to live with the inherent complexity of the rule, but let′s not introduce complexity unless it is absolutely necessary. A rule of this type could be the computation of automobile insurance premiums, which depends on attributes the car (one object) and the attributes of the owner (another object).

Summary of Associations

This list shows the 15 associations (10 if you count distinct names) and the 5 objects (6 if you add the minor object of user role). VisualAge Requirements Tool can produce very useful views with this small set of elements.

User Role Diagram

The User Role diagram provides what some methodologies call the *context diagram*, in which the system under consideration is shown connected with its external agents or actors to indicate interchanges (transactions) between them.

This diagram allows the interaction between the analyst specifying an operation and business users. Each type of user should concentrate on his/her own transactions and establish if the list is complete. Are there more functions required? Remember, that we want to make sure that all transactions are recorded - not surprises when the system is installed.

Business Transaction Flow Diagram

Transaction flow diagrams make most sense when displayed for input transactions. They express the full behavior of a system function.

We can safely say that the totality of the functions of an operation is expressed by the aggregation of all of its transaction flows.

Transaction flows relate directly to implementation elements, as VRT does with VisualGen*, because they represent the known paradigm of INPUT-PROCESS-OUTPUT. A very simple way of describing function in a (almost) non-procedural way.

Transaction flows represent process steps in a business process view. They have inputs and outputs that connect to prior or subsequent steps in the business process. They are the element of connectivity with tools like ProModeler*.

Consolidate Data

This step is essentially a review of the specification as its stands by applying the ideas mentioned before:

- Are the names given to business elements clear?
- Have we missed transactions?
- Have we properly defined information and dependencies?
- Have we discovered all static rules?
- Have we discovered all dynamic rules?
- Are the user functions provided by transaction flows correct?
- Etc.

Review Your Work with Your Users

Users are needed to validate the functions specified in the business operation.

- Does every transaction have all the needed facts to be useful?
- Do dynamic rules consider all exception conditions?
- Are there business object sub-classes that need to be considered (and information elements broken down)?
- Do we have all rules that the operation needs for each business object?
- Etc.

Team-based Applications Development

VisualAge Requirements Tool is a tool that utilizes TeamConnection* as its repository in the Professional edition. The Standard edition uses ObjectStore** (by Object Design Inc.).

Team Connection offers a variety of control functions that enable the sharing of information by various tools. The VRT data is accessible to any tool integrated on TeamConnection. The tools currently operating on the TeamConnection base are: VisualAge Requirements Tool, DataAtlas* and VisualGen.

Scenario: Integration with VisualGen

VisualAge Requirements Tool has specific features to enable the mapping of requirements elements to VisualGen's implementation elements.

One can indicate what VisualGen element implements each of the VRT objects. The user can look at a business rule (for example) and determine what VisualGen *called batch application* implements it. The user working on a new requirement can analyze the change from the business side and find out what elements are impacted on the implementation side. Conversely, for each implementation element one can find out which is the requirement object that it implements.

The transaction flow diagram provides a powerful way of visualizing VisualGen function. Its input-process-output looking format offers direct implementation guidance. The rule re-use facilities in VRT can help determine that some rule is already implemented and hence can be re-used without requiring new code.

VisualAge Requirements Tool to VisualGen Mappings

The information in this foil does not need explanation.

Process Tools and VisualAge Requirements Tool

As a complement to the information in on the foil we can say:

Business user understand operational aspects of an enterprise using two mental models: the *business process* and the *business operation*.

The business process view is similar to the view of a manufacturing line, well known because it appears often when describing procedures. In manufacturing, the industrial engineer specializes in the design and evaluation of these lines and worries about *traffic* and *quality* related issues. He/she is concerned with capacities, wait-times, elapsed-times, resources consumed, etc. The objective of the analysis is to optimize use of resources, improve quality, and produce quickly.

The business operation view is similar to the view of a single manufacturing process station. In manufacturing, the process engineer is in charge of defining and optimizing the performance of a single work-station - a process step - regardless of its position in the manufacturing line. The process engineer is concerned with the behavior - *functions*, *performance*, and *data requirements* - of his/her machine.

To summarize, the concept of business operation in VRT is: a group of business functions representing *processes steps* to be used in one or more business processes.

Scenario: Usage with ProModeler

Once the "to-be" processes are decided upon in a business process re-engineering (BPR) project, someone is assigned the responsibility to implement them. The implementation of the new business process will involve, among many other things, the creation of facilities to carry out each of the *required business functions* that are shown as process steps in business process diagrams. These facilities can be manual or automated and they may exist, may require reworking existing ones, or may require a totally new implementation. Typically, the IT architect is in charge of deciding how to develop the automated facilities and the user analyst in charge of deciding the operative changes for the departments responsible for manual business functions.

The architect can look at each required business function alternatively, as a step in a business process or as a transaction flow in the business operation that will implement it. Hence, the architect has a major sorting task: he/she has to assign each automated process step to an application system (business operation in VRT terms). Once the assignment is made, business and IT analysts can start the detailed definitions of transactions, facts, information and business rules.

Notice that the VRT environment enables and focuses on aspects of the transaction flows that were not considered in the BPR focus. Some of these are:

- Full description of facts (terminology) in a dictionary
- Detailed description of each transaction, its facts, and its static rules
- Detailed description of each information, its facts, and its static rules
- Specification of the rules of behavior of each transaction flow (required business function).

Also notice that the view of a group of functions as a system, as supported by VRT, brings in additional focus on some of the functions required to maintain the system operational. For example it is very doubtful that a customer change of address transaction will be listed while doing BPR on the order entry process. This is because the maintenance type functions seen from a system view tend to be overhead, just to keep the information current, and not provide a pro-active business edge.

The data that needs to be transferred from ProModeler in the scenario mentioned above is rather small. Probably the creation of some business operations, and for each, a set of transactions that are involved in the process steps assigned to that business operation. Potentially, some of the business facts will also be available. Business rules describing the behavior of transactions will not be available.

Import/Export Facilities

VRT′s External File Format (EFF) is CDIF compliant and offers a wide range of possibilities for integration with tools not in TeamConnection.

- ;**Export and import from any VRT to any VRT** - this is useful for the consolidation of work done by different analysts working in parallel and for migration between different versions of VRT, if incompatible changes are needed. This last aspect assures users that the result of their work will not be invalidated by future versions of the tool.
- **Export and import from other tools** - this is intended for users that either have some of the data available from previous work on other tools and want this data in VRT, or that want to pass-on the VRT data to other tools of their choice. These transfers can be achieved by writing a program that produces a file matching the input format of the receiving tool by reformatting the export file of the sending tool.
- **Storage of the VRT data in other databases** - this allows users to store the VRT data in any database. The DBMS can manipulate the VRT data or produced special purpose reports. Since the EFF contains every piece of data in the VRT specification, no data is lost by exporting it.

Standard and Professional Editions

The Standard edition is intended for consultants that work alone and analysts that need to travel to perform their duties. This edition has exactly the same function as the Professional edition, except for the versioning controls offered by TeamConnection. Information from the Standard edition can be exported into the Professional edition and vice-versa.

The Professional edition offers a more complete environment for group activities. It benefits by sharing information with TeamConnection enabled tools and by allowing versioning controls.

Engages the Business Community

Business users are the drivers of business change and VRT allows them to be included in the requirements process in a simple way. There are no steep learning curves and the benefits of solid terminology and crisp presentations can be attained with a small effort.

The gains achieved by involving the business user in the requirements process are considerable. They range from good user/development relationships to successfully implemented systems, which allow the business to evolve and compete.

Engages the Development Community

Developers will have the assurance of a well thought out set of requirements, they will have a ready-reference to what users want, and can maintain the connection between requirements and their implementation objects. This leads to quicker reaction to user′s needs and the satisfaction of knowing that they have control over their development and the future evolution of their systems.

From VRT they obtain information directly usable for development projects:

- Identification and definition of data requirements
- Identification and definition of all inputs and outputs
- Identification and definition of processing rules

They can also derive function point estimates at the beginning of the project and have firm data for discussions involving how to phase-in the project matching resources available.

Developers can use the VRT information to develop plans for the functional acceptance test of their product. They know business users will like to test every transaction and every business rule to assure that the correct function is delivered.

Summary

Here is the time for the speaker to relate some personal story about failed development projects and transmit the idea that solid requirements is a necessary ingredient of successful projects.

The main distinguishing factor of VisualAge Requirements Tool is that it supports business users in the requirements process. If business users can communicate with each other and IT, they all gain by creating systems that truly support required business functions.