

A Standards-Based Approach to Extracting Business Rules

Ira Baxter
Semantic Designs, Inc.

Stan Hendryx
Hendryx & Associates



Who are the presenters?

- Semantic Designs

Architecture-Driven Modernization (ADM)

- Automated Analysis and Enhancement of Large Scale Software Systems
 - DMS: generalized compiler technology for effecting massive change
 - Supplier of COTS Software Engineering Tools: DMS, Test Coverage, Metrics, ...
- Ira D. Baxter, Ph.D., CEO, Semantic Designs
 - 35 years Software Engineering R&D:
Operating Systems, Compilers, Transformation Systems, Reuse
 - CoChair of International Conference on Software Maintenance 2002
 - Architect of Design Maintenance System (DMS)

- Hendryx & Associates

Semantics of Business Vocabulary and Business Rules (SBVR)

- Consulting and Services to IT Management
- Stan Hendryx, MS MIT, CEO, Hendryx & Associates
 - 20 Years Information Technology Management Consultant
 - Co-chairman, OMG Business Modeling & Integration Task Force
 - Co-submitter, OMG Semantics of Business Vocabulary and Business Rules
 - Strategic Advisory Board, European Digital Business Ecosystem Project

Problem Context for Business Rule Extraction

- What's the whole problem?
- What are business rules?
- Tools to extract rules

- Legacy system needs ongoing enhancement
 - *Success* breeds discontent!
 - Desired functionality changes or improvements
 - Must integrate with other systems
 - Existing Business logic is critical
 - But software is complex and poorly documented
 - Business Rules are hidden in the code
- Reliable and effective change requires
 - *Extraction of explicit business rules from the software*
 - Traceability of business rules to implementing software
 - Analysis of business rules for continued relevance
 - Alignment with business rules of other organizations

Architecture-Driven Modernization (ADM)

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Proposal for Business Rule Extraction

- Use most advanced business rule concepts available:
 - Semantics of Business Vocabulary and Business Rules (SBVR)**
 - Emerging from industry thought leaders
 - based on formal logic and linguistics
 - Open, industry standard format
 - interchangeable and machine-processable by different tools
- Use Compiler-grade analysis of source code
 - Yield high degrees of automation
 - Provide maximum accuracy by providing access to code details
- Enable Business Analysts to understand/document Business Rules
 - Present source code to BA with *code-analysis and BA annotations*
 - BA navigates code quickly to enhance understanding
 - *Capture BRs as SBVR*, and capture BR connections to code



Not all rules from code are Business Rules

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Typical Problem (Rule #s):

- Nonsensical rule (1)
- COBOL symbols are not business terms (1, 4, 6)
- Direct use of implementation technology is not business vocabulary or business rule (1, 2, 3, 4, 5, 6, 7, 12)
- Over specification (2, 3)
- Failure to abstract variable (“Inventory Cat”) to named business term (8, 10, 11)
- Duplicated (cloned) business terms implies independence when rules may be coupled (4&7, 8&9)

From a “BR tool” vendor’s Web site

- (1) End of File Switch Is Equal TO 1 Apply VALIDATE-PUR-MASTER Until End Of File Switch Is Equal To 1
- (2) 1 Is Added To Error Counter
When Purchase Order Vendor Number Is Equal To Spaces Or Purchase Order Vendor Number Is Equal To Zero
- (3) Error Message Is Equal To ‘VENDOR NUMBER MAY NOT BE NULL’
When Purchase Order Vendor Number Is Equal To Spaces Or Purchase Order Vendor Number Is Equal To 0
- (4) Working Area For Quantity Due Is Computed As PUR-ORD-QTY Minus PUR-REC-QTY Plus Purchase Order Quantity Returned To Vendor
When Purchase Order Status Is Equal To ‘O’
- (5) 1 Is Added To Error Counter When Working Area For Quantity Due Is Equal To 0
- (6) Error Message Is Equal to ‘OPEN ORDER HAS ZERO BALANCE DUE’
When Working Area For Quantity Due Is Equal To 0
- (7) Working Area For Quantity Due Is Computed As PUR-ORD-QTY Minus PUR-REC-QTY Plus Purchase Order Quantity Returned To Vendor
When Purchase Order Status Is Equal To ‘C’
- (8) Inventory Cat Is Equal To 86
When Inventory Purchase Order Cost Is More Than 500.00 And Inventory Purchase Order Cost Is Less Than 1000.00
- (9) Inventory Sales Price Is Computed as Inventory Purchase Order Cost Multiplied By Inventory Markup Percent
When Inventory Purchase Order Cost Is More Than 500.00 And Inventory Purchase Order Cost Is Less Than 1000.00
- (10) Inventory Cat Is Equal To 87
When Inventory Purchase Order Cost Is More Than 1000.00 And Inventory Purchase Order Cost Is Less Than 5000.00
- (11) Inventory Cat Is Equal To 90
When Inventory Purchase Order Cost Is More Than 5000.00
- (12) End of File Switch Is Equal To 1



- What's the whole problem?
- What are business rules?
- Tools to extract rules

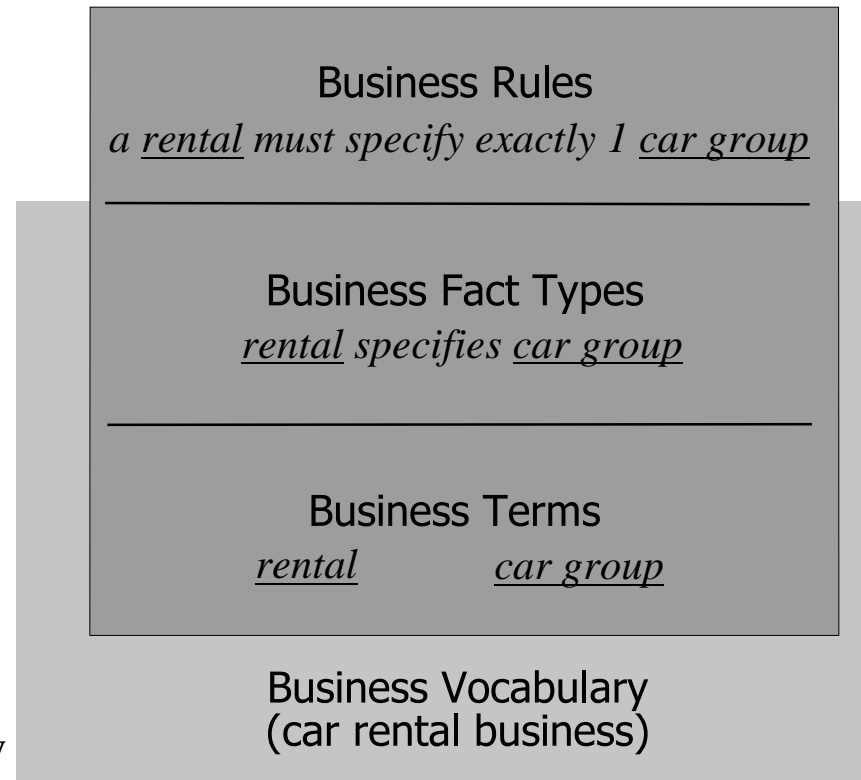
What is a Business Rule?

- Actionable business directive whose purpose is to advise or inform and that introduces ...
 - an *obligation* that covers conduct, action, practice, or procedure, or
 - a *necessity* that is intended as a definitional criterion.
- A business rule is typically expressed in an English declarative statement (not as a process or procedure).
 - tabular forms are also common
- Enforcement of a business rule is separate from the rule itself
 - partly enforced by daily, manual business activities
 - partly enforced by IT systems
 - may span multiple systems
 - system requirements are rules about enforcing business rules in the system

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Business Vocabulary and Business Rules

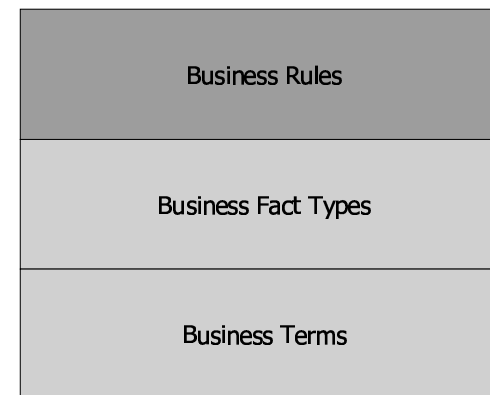
- Business rules build on business fact types.
- Business fact types build on business terms.
- A business vocabulary comprises business terms and business fact types.
- To get business rules:
 - First get the business vocabulary
 - Then build rules using vocabulary
 - Interleave activities in practice



- What's the whole problem?
- What are business rules?
- Tools to extract rules

Business Vocabulary and Business Rules Are Independent Of Implementation

- Not dependent on any business process, information system, or record keeping system.
- Business rules depend only on the business vocabulary.
 - Being implementation independent, a vocabulary and rules set can inform or validate any implementation design or integration of implementations.
 - Being in English, a vocabulary and rules set can be validated and maintained by business people.



- What's the whole problem?
- What are business rules?
- Tools to extract rules

The Code only *hints* at Business Rules

- Code (COBOL, Database Schemas, JCL, Screen Definitions...)
 - Contains
 - data definitions, information flows, computational procedures, comments
 - *Represents*
 - Business vocabulary
 - Fragments of Business rule enforcement procedures
 - Software architecture vocabulary, structures, procedures
 - Hardware and communication vocabulary, procedures
- Business meaning of the data and actions
 - is determined only at the system inputs and outputs.
- ➔ Business rules are usually *not* in the code
 - Information is lost or tangled when programmed
 - Partly in organization context of software
 - *Requires people to recover (induce) this information using code clues*
- Automated extraction of business rules is often proposed
 - Can at best be heuristic
 - Defects: missing rules, incorrect rules, ...

- What's the whole problem?
- What are business rules?
- Tools to extract rules

How to Extract Business *Vocabulary*

- Extraction requires *interactive process with business analysts*
 - Using code clues in the code
 - Using analyst's understanding of the business context of system
- Major clues to business vocabulary in the code
 - Program symbolic names and types – data, functions, arguments
 - System-wide program data flows and visibility of variables
 - Equivalences of various code concepts
 - Labels on input/output forms and in report generator programs
 - Program comments
- Code analysis tools can extract these clues
 - Need compiler-level detail across application languages
- Analysis results can be presented to business analysts
 - who apply business judgment to formulate business vocabulary in English
 - and record vocabulary and code connections

- What's the whole problem?
- What are business rules?
- Tools to extract rules

How to Extract Business *Rules*

- Extraction requires interactive process
- Major clues to business rules in the code:
 - Code fragments representing business vocabulary
 - System-wide program logic and control flow
 - How code concepts are combined to produce results
 - Program operations and functions
 - Program execution environment
 - i.e. job control code, batch schedules, interactive processing
 - User error messages
 - Program comments
- Business Analyst needs analysis-based code browser to identifies fragments for BRs, then writes BR in English
 - Selecting fragment selects corresponding business terms
 - BA writes BR using SBVR (“Semantics of Business Vocabulary and Rules”)
 - Written rule is checked for syntax and proper use of business terms
 - Written rule is tied to program fragment for traceability

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Extracted Vocabulary and Rules Inherently Traceable To and From Code

- System-wide data flows and reaches of each program symbol are correlated with a vocabulary entry that may correspond to the symbol.
- Program code corresponding to each idiom together with the types of the variables and functions in the idiom correlate to a business rule.
- A database will be maintained of the links between the vocabulary and rule base, and the code base.

Extracting Rules from COBOL

- What's the whole problem?
- What are business rules?
- Tools to extract rules

COBOL Code

```
MOVE 1000 TO MIN-PAYMENT-ALLOWED.  
COMPUTE CURRENT-PAYMENT = (NEW-CURR-BALANCE * 0.1).  
IF CURRENT-PAYMENT < MIN-PAYMENT-ALLOWED AND NEW-CURR-BALANCE > 1000 THEN  
    MOVE MIN-PAYMENT-ALLOWED TO CURRENT-PAYMENT.  
IF NEW-CURR-BALANCE < 1000 THEN  
    MOVE NEW-CURR-BALANCE TO CURRENT-PAYMENT.  
IF NEW-CURR-BALANCE < 1 THEN  
    MOVE ZEROES TO CURRENT-PAYMENT.
```

current payment

10%

current balance

current balance

\$10.00

Individual Rules

- if current balance is \$100.00 or more then current payment is 10% of current balance.
- if current balance is less than \$100.00 and is \$10.00 or more then current payment is \$10.00.
- if current balance is less than \$10.00 and is \$0.01 or more then current payment is current balance.
- if current balance is less than \$0.01 then current payment is \$0.00.

Composed Rule (decision table)

current payment is

determined by current balance

- if \$100.00 or more then 10% of current balance
- else if less than \$100.00 and is \$10.00 or more then \$10.00
- else if less than \$10.00 and is \$0.01 or more then current balance
- else if is less than \$0.01 then \$0.00.

- What's the whole problem?
- What are business rules?
- Tools to extract rules

SBVR: An Open Standard for Business Vocabularies and Business Rules

- SBVR is an emerging OMG standard.
 - Expected to be adopted by OMG in November
- SBVR vocabularies and rules
 - can be stored in any MOF-compliant repository.
 - are formally represented in OMG standard MOF/XMI format.
 - will be usable with OMG's Model Driven Architecture™ (MDA).
- Other commercial and open source SBVR tools are forthcoming, from multiple vendors.
- Industry associations will develop SBVR vocabularies and rules
 - promotes business interoperability
- SBVR selected for the Business Modeling Language for the open source European Digital Business Ecosystem.

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Potential uses of Extracted Business Vocabularies and Business Rules

- Understand function of current system in organization's terms
- Maintain in English by organization's business analysts.
- Integrate legacy vocabulary and rules with new vocabulary and rules.
- Automate transformation/generation of many software components at interfaces between systems.
- Reuse in other applications and departments that use the same vocabulary and rules.
- Train new personnel on the vocabulary and rules.
- Support audits for regulatory compliance.
- Develop system requirements, design validation, and acceptance test specifications for systems based on the vocabularies and rules.

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Business Rule Expression Process and Tools

Iterative, Interactive, Systematic Team Process

1. Find where functionality is implemented in code
 - *Software Test Coverage Tool*
2. Determine business vocabulary symbolized in code
 - *System-wide Information Flow Tool*
 - *Business Vocabulary Assistant Tool (Browser/Annotation)*
 - *E-R Extraction Tool*
3. Discover business rules enforced by code
 - *Business Rule Assistant (Browser/Annotation) Tool*
 - *Clone Detection Tool*

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Business Rule Expression Overview

1. Enable Business Analysts (BAs) to browse/understand code

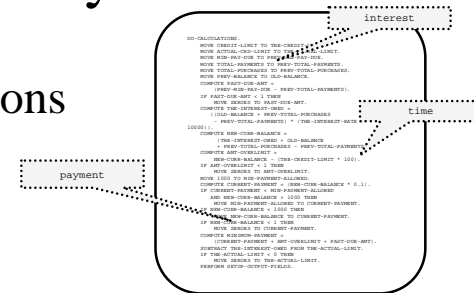
- Full System Browsers for *multiple languages* in system
- System Wide Information flow trace
- Entity-Relationship extraction tool
- Test Coverage

```

00-CALCULATIONS:
NOW CREDIT-LIMIT TO THE CREDIT-LIMIT
NOW ACTUAL-CRD-LIMIT TO THE ACTUAL-LIMIT
NOW MIN-PAY-DUE TO PREY-PAY-DUE
NOW TOTAL-PAYMENTS TO PREY-TOTAL-PAYMENTS
NOW TOTAL-PAYMENTS TO PREY-TOTAL-PAYMENTS
NOW PREY-BALANCE TO OLD-BALANCE
COMPUTE PREY-CRD-LIMIT
IF PREY-CRD-LIMIT < PREY-TOTAL-PAYMENTS
  THEN PREY-CRD-LIMIT = PREY-TOTAL-PAYMENTS
COMPUTE THE-INTEREST-RATE =
  (OLD-BALANCE * PREY-TOTAL-PAYMENTS
  / (PREY-TOTAL-PAYMENTS) * (THE-INTEREST-RATE /
  100))
COMPUTE NEW-CRD-BALANCE =
  (THE-INTEREST-RATE * OLD-BALANCE
  + PREY-TOTAL-PAYMENTS - PREY-TOTAL-PAYMENTS)
COMPUTE NEW-CRD-LIMIT
IF NEW-CRD-BALANCE < (THE-CREDIT-LIMIT * 100)
  THEN NEW-CRD-LIMIT = THE-
  CREDIT-LIMIT
IF NEW-CRD-BALANCE < 0 THEN
  NOW SEND TO ANT-CRD-LIMIT
COMPUTE CURRENT-PAYMENT = (NEW-CRD-BALANCE * 0.1)
IF CURRENT-PAYMENT < NEW-PAYMENT-ALLOWED
  THEN NEW-PAYMENT-ALLOWED = CURRENT-PAYMENT
IF NEW-CRD-BALANCE < 0 THEN THEN
  NOW NEW-PAYMENT-ALLOWED TO CURRENT-PAYMENT
  NOW SEND TO CURRENT-PAYMENT
  NOW SEND TO CURRENT-PAYMENT
COMPUTE MINIMUM-PAYMENT =
  CURRENT-PAYMENT * ANT-CRD-LIMIT + PART-001-AMT
  DEBITANT THE-INTEREST-CHRG FROM THE-ACTUAL-LIMIT
  IF THE-ACTUAL-LIMIT < 0 THEN
    NOW SEND TO THE-ACTUAL-LIMIT
    PERFORM STEP-OUTPUT-FIELDS
  
```

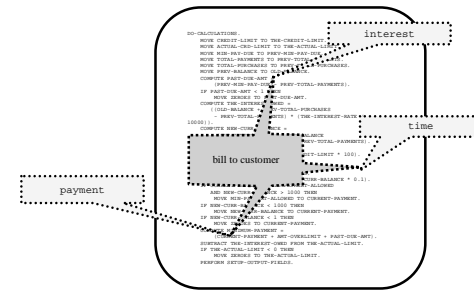
2. BAs annotate source code with Business Vocabulary & Terms

- Builds up Business Vocabulary
- Use understanding to determine what code does what functions
- Ties source code elements to Business Rule Elements



3. BAs assemble annotations into Business Rules

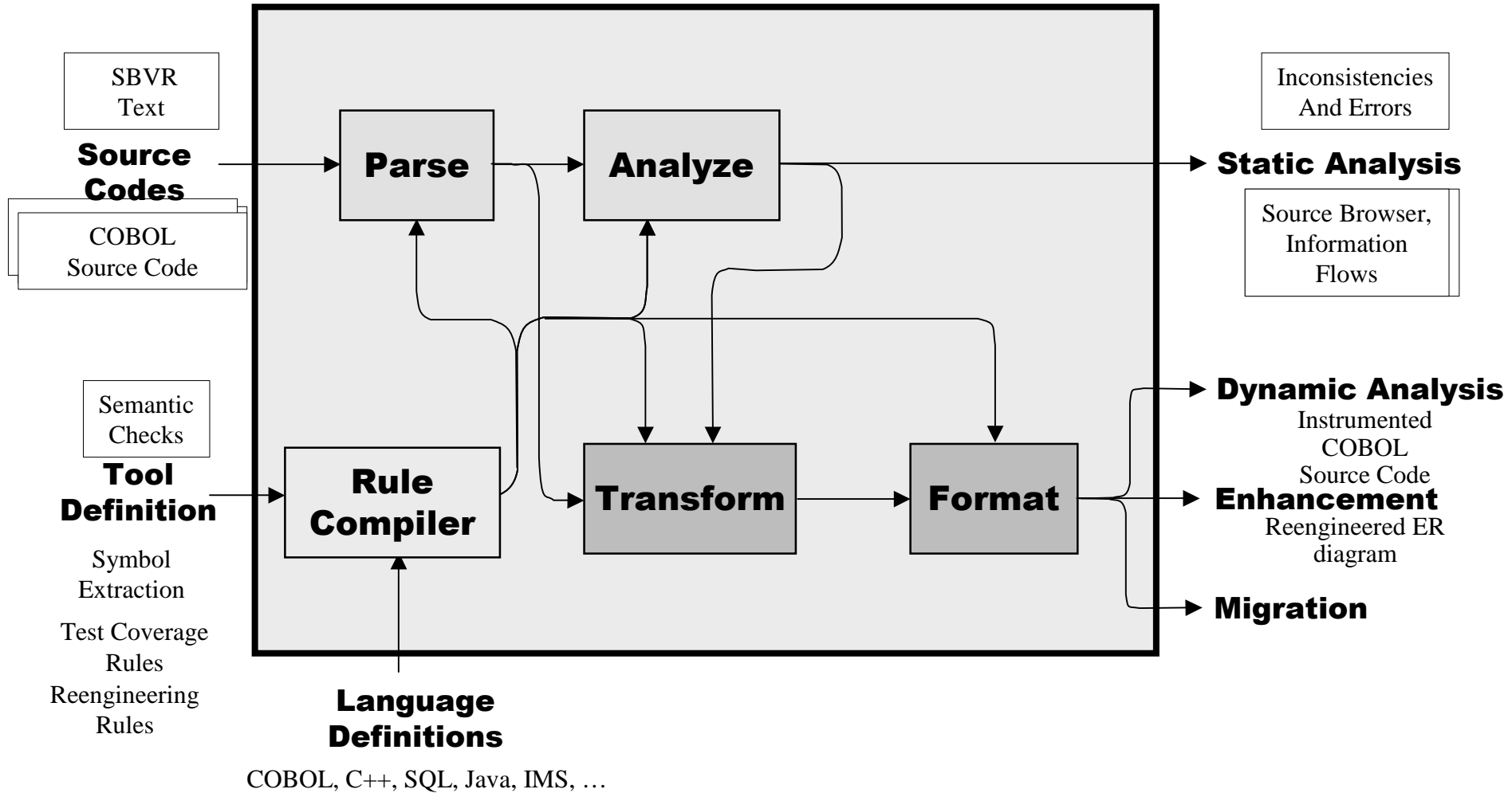
- Formal encoding in SBVR
- Automatic checking for errors in rules
- English Paraphrase to aid understanding
- Traceability from SBVR back to source code



How A Generalized Compiler Works

Examples For Typical BR Extraction

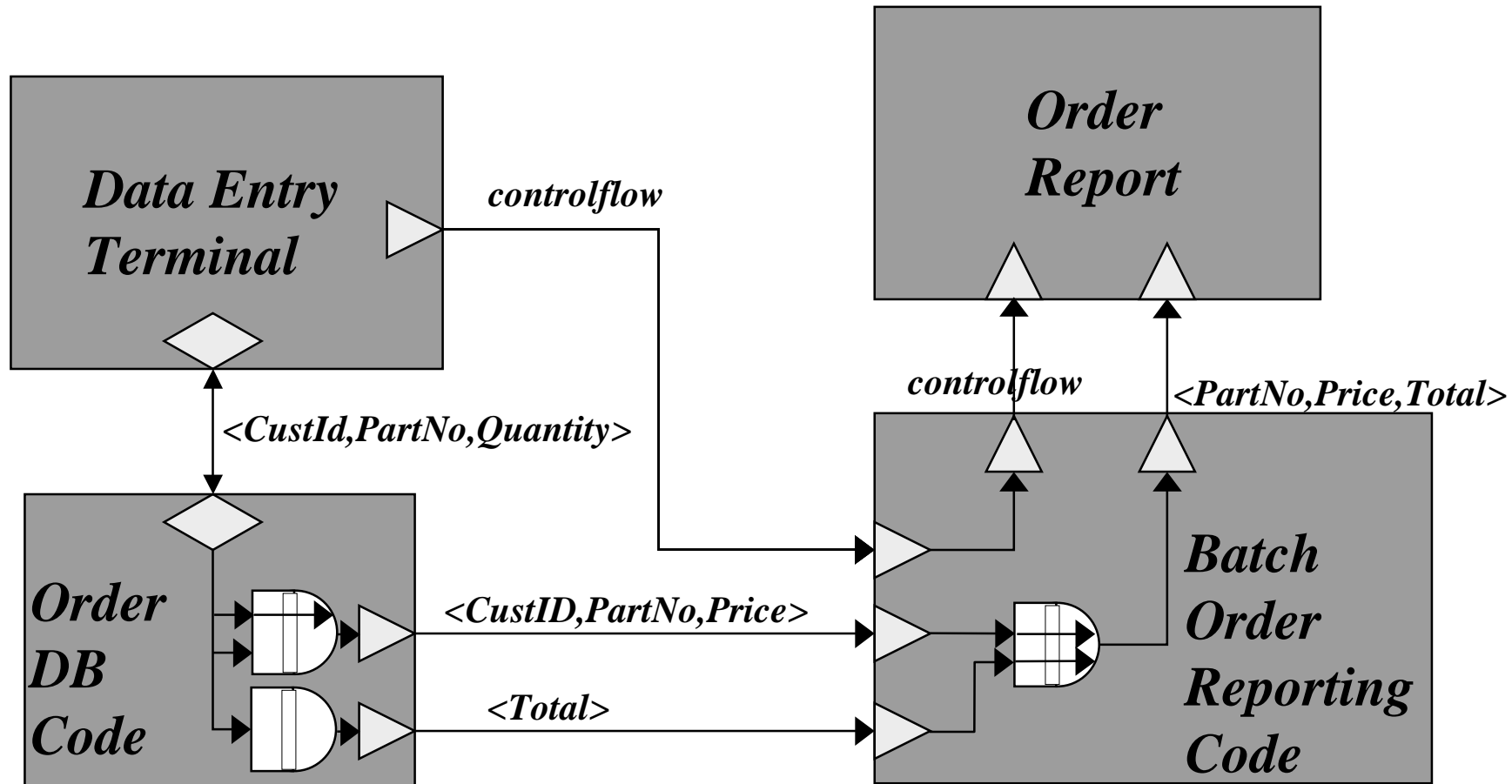
- What's the whole problem?
- What are business rules?
- Tools to extract rules



- What's the whole problem?
- What are business rules?
- Tools to extract rules

System Wide Information Flow Tool

Shows Control and Data Flow: “What gets where?”



Composes information from multiple languages: COBOL, DB, Screens, ...

Software Test Coverage

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Used to find where requirements are implemented

- Discovery of code executed by test cases
 - Non-executed code likely to be dead or flawed
 - Normally used for software quality assessment
- Our purpose: finding features in code
 - *Code exercised by test case is related to feature exercised by test case!*
- Process
 - Instrument program to track what it executes
 - Execute a test case that exercises feature of interest
 - Display executed part of program: must contain feature
- How it works: tracking program control flow
 - Break program into conditionally-executed blocks (paragraph, IF)
 - Put probe into block to remember block was “executed”
 - Display execution status of program blocks

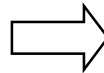
Software Test Coverage Tool

Finds business functionality in code

- What's the whole problem?
- What are business rules?
- Tools to extract rules

COBOL program...

```
DO-CALCULATIONS.  
MOVE CREDIT-LIMIT TO THE-CREDIT-LIMIT.  
MOVE ACTUAL-CRD-LIMIT TO THE-ACTUAL-LIMIT.  
MOVE MIN-PAY-DUE TO PREV-MIN-PAY-DUE.  
MOVE TOTAL-PAYMENTS TO PREV-TOTAL-PAYMENTS.  
MOVE TOTAL-PURCHASES TO PREV-TOTAL-PURCHASES.  
MOVE PREV-BALANCE TO OLD-BALANCE.  
COMPUTE PAST-DUE-AMT =  
  (PREV-MIN-PAY-DUE - PREV-TOTAL-PAYMENTS).  
IF PAST-DUE-AMT < 1 THEN  
  MOVE ZEROES TO PAST-DUE-AMT.  
COMPUTE THE-INTEREST-OWED =  
  ((OLD-BALANCE + PREV-TOTAL-PURCHASES  
  - PREV-TOTAL-PAYMENTS) * (THE-INTEREST-RATE / 10000)).  
COMPUTE NEW-CURR-BALANCE =  
  (THE-INTEREST-OWED + OLD-BALANCE  
  + PREV-TOTAL-PURCHASES - PREV-TOTAL-PAYMENTS).  
COMPUTE AMT-OVERLIMIT =  
  NEW-CURR-BALANCE - (THE-CREDIT-LIMIT * 100).  
IF AMT-OVERLIMIT < 1 THEN  
  MOVE ZEROES TO AMT-OVERLIMIT.  
MOVE 1000 TO MIN-PAYMENT-ALLOWED.  
COMPUTE CURRENT-PAYMENT = (NEW-CURR-BALANCE * 0.1).  
IF CURRENT-PAYMENT < MIN-PAYMENT-ALLOWED  
  AND NEW-CURR-BALANCE > 1000 THEN  
  MOVE MIN-PAYMENT-ALLOWED TO CURRENT-PAYMENT.  
IF NEW-CURR-BALANCE < 1000 THEN  
  MOVE NEW-CURR-BALANCE TO CURRENT-PAYMENT.  
IF NEW-CURR-BALANCE < 1 THEN  
  MOVE ZEROES TO CURRENT-PAYMENT.  
COMPUTE MINIMUM-PAYMENT =  
  (CURRENT-PAYMENT + AMT-OVERLIMIT + PAST-DUE-AMT).  
SUBTRACT THE-INTEREST-OWED FROM THE-ACTUAL-LIMIT.  
IF THE-ACTUAL-LIMIT < 0 THEN  
  MOVE ZEROES TO THE-ACTUAL-LIMIT.  
PERFORM SETUP-OUTPUT-FIELDS.
```



... with probes added

```
DO-CALCULATIONS.  
MOVE 1 TO PROBE(1).  
MOVE CREDIT-LIMIT TO THE-CREDIT-LIMIT.  
MOVE ACTUAL-CRD-LIMIT TO THE-ACTUAL-LIMIT.  
MOVE MIN-PAY-DUE TO PREV-MIN-PAY-DUE.  
MOVE TOTAL-PAYMENTS TO PREV-TOTAL-PAYMENTS.  
MOVE TOTAL-PURCHASES TO PREV-TOTAL-PURCHASES.  
MOVE PREV-BALANCE TO OLD-BALANCE.  
COMPUTE PAST-DUE-AMT =  
  (PREV-MIN-PAY-DUE - PREV-TOTAL-PAYMENTS).  
IF PAST-DUE-AMT < 1 THEN  
  MOVE 1 TO PROBE(2).  
  MOVE ZEROES TO PAST-DUE-AMT.  
COMPUTE THE-INTEREST-OWED =  
  ((OLD-BALANCE + PREV-TOTAL-PURCHASES  
  - PREV-TOTAL-PAYMENTS) * (THE-INTEREST-RATE / 10000)).  
COMPUTE NEW-CURR-BALANCE =  
  (THE-INTEREST-OWED + OLD-BALANCE  
  + PREV-TOTAL-PURCHASES - PREV-TOTAL-PAYMENTS).  
COMPUTE AMT-OVERLIMIT =  
  NEW-CURR-BALANCE - (THE-CREDIT-LIMIT * 100).  
IF AMT-OVERLIMIT < 1 THEN  
  MOVE 1 TO PROBE(3).  
  MOVE ZEROES TO AMT-OVERLIMIT.  
MOVE 1000 TO MIN-PAYMENT-ALLOWED.  
COMPUTE CURRENT-PAYMENT = (NEW-CURR-BALANCE * 0.1).  
IF CURRENT-PAYMENT < MIN-PAYMENT-ALLOWED  
  AND NEW-CURR-BALANCE > 1000 THEN  
  MOVE 1 TO PROBE(4).  
  MOVE MIN-PAYMENT-ALLOWED TO CURRENT-PAYMENT.  
IF NEW-CURR-BALANCE < 1000 THEN  
  MOVE 1 TO PROBE(5).  
  MOVE NEW-CURR-BALANCE TO CURRENT-PAYMENT.  
IF NEW-CURR-BALANCE < 1 THEN  
  MOVE 1 TO PROBE(6).  
  MOVE ZEROES TO CURRENT-PAYMENT.  
COMPUTE MINIMUM-PAYMENT =  
  (CURRENT-PAYMENT + AMT-OVERLIMIT + PAST-DUE-AMT).  
SUBTRACT THE-INTEREST-OWED FROM THE-ACTUAL-LIMIT.  
IF THE-ACTUAL-LIMIT < 0 THEN  
  MOVE 1 TO PROBE(7).  
  MOVE ZEROES TO THE-ACTUAL-LIMIT.  
PERFORM SETUP-OUTPUT-FIELDS.
```

Test Coverage Display Tool

- What's the whole problem?
- What are business rules?
- Tools to extract rules

Shows what was executed when searching for a feature

The screenshot shows the 'Test Coverage Vectors Display' application. The interface is divided into several sections:

- Menu:** Open PRF, Open TCVs, Save TCVs, Exit, About.
- Unselected TCVs:** An empty list box.
- Select/Unselect:** Buttons for Select, Unselect, Select All, Unselect All, Remove TCVs, and Report.
- Selected TCVs:** A list box containing 'S:\Customers\GeneralServicesAdministration\GSADemo\GSA-test-coverage-demo.tcv'.
- File Explorer:** Shows a tree view of files and probes. The 'All Files/Probes' tab is active. The tree shows a folder structure with files like '50Ex.COB', 'CCVS85.co', and 'billing.cob'. 'Line 452' is highlighted in the tree.
- Code Editor:** Displays the contents of 'billing.cob' starting at 'Line 452/813'. The code includes several lines of COBOL logic, such as 'DO-CALCULATIONS.', 'MOVE CREDIT-LIMIT TO THE-CREDIT-LIMIT.', and 'COMPUTE AMT-OVERLIMIT = NEW-CURR-BALANCE - (THE-CREDIT-LIMIT * 100).'. Line 452 is highlighted in the editor.
- Status Bar:** Shows 'Total Probes: 94', 'Covered: 28(29.7%)', and 'Uncovered: 66(70.2%)'.



- What's the whole problem?
- What are business rules?
- Tools to extract rules

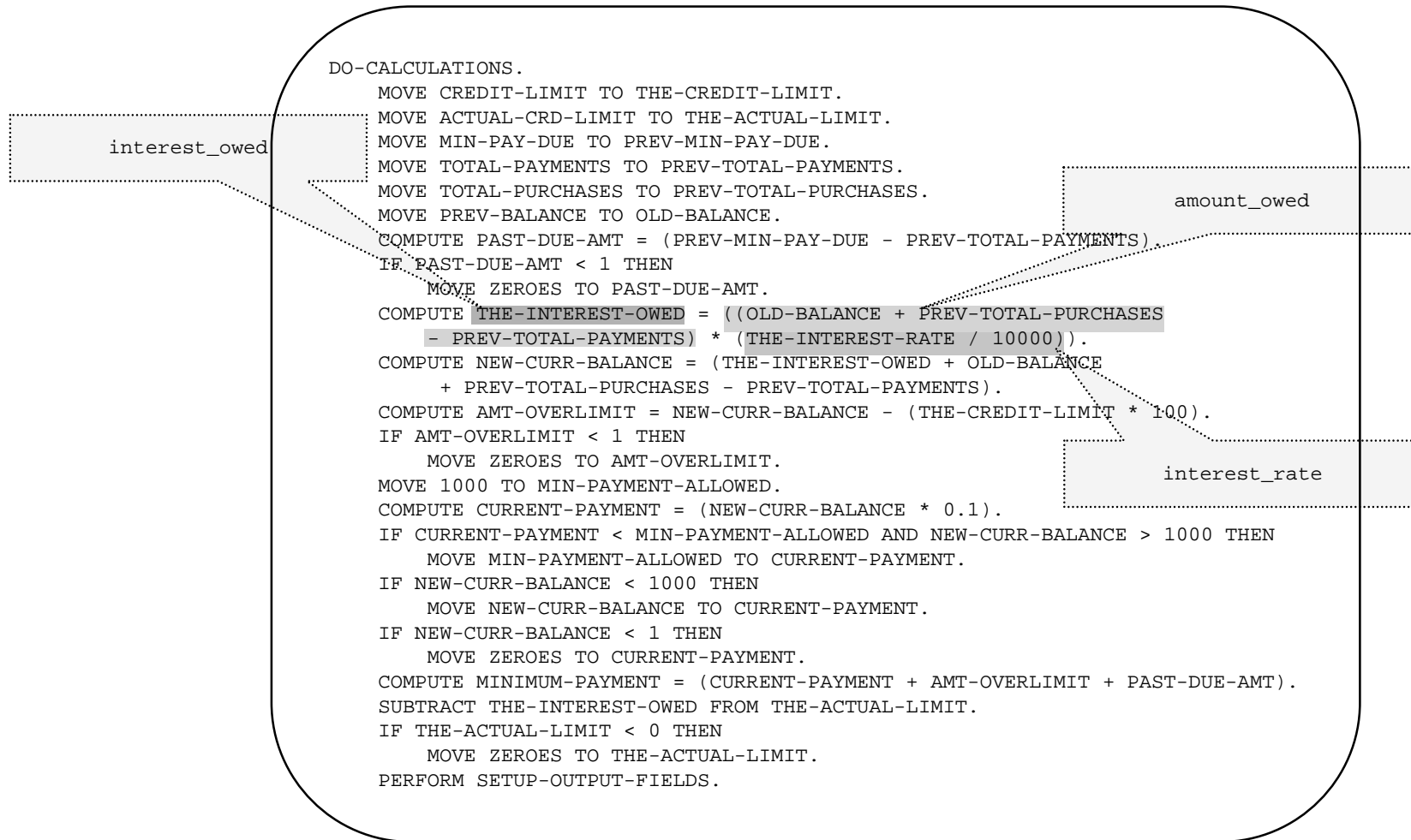
Annotation of Source Code with Business Concepts

- Business Analysts review code
 - Using Test Coverage, Clone detection to find concepts
 - Using Source Browser, annotate code
 - Using Business Rules Concept names
 - Using Business Rule Fragments
 - Using arbitrary text
- Annotated code available to all Business Analysts
 - Helps build shared vocabulary, rule fragments
- Annotated code given to Business Rule Assistant
 - Uses annotations and data flows to extract rules

Source Browser and Annotation Tool

- What's the whole problem?
- What are business rules?
 - Tools to extract rules

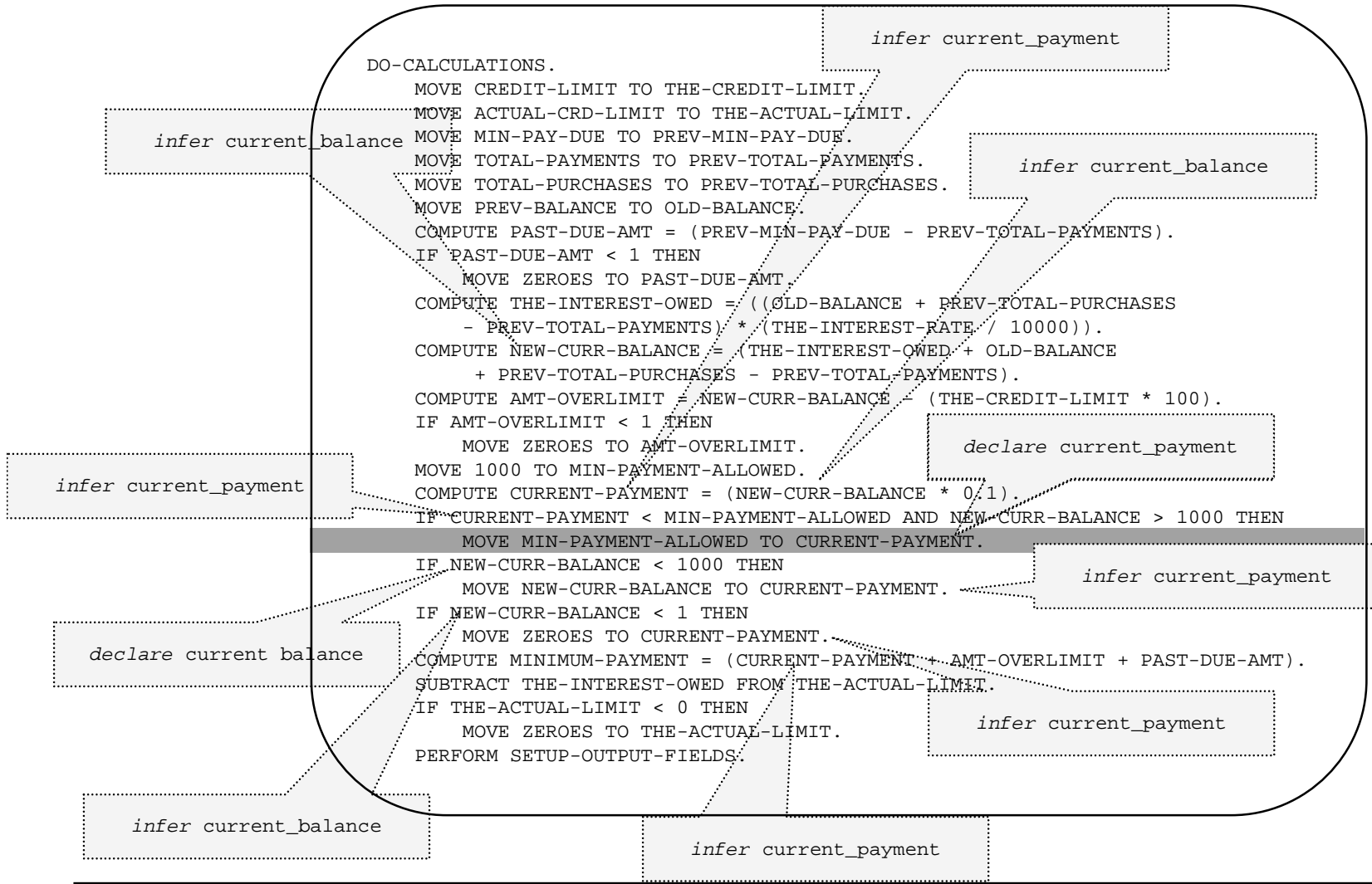
1) Annotate source code syntactic entities



Source Browser and Annotation Tool

- What's the whole problem?
- What are business rules?
- Tools to extract rules

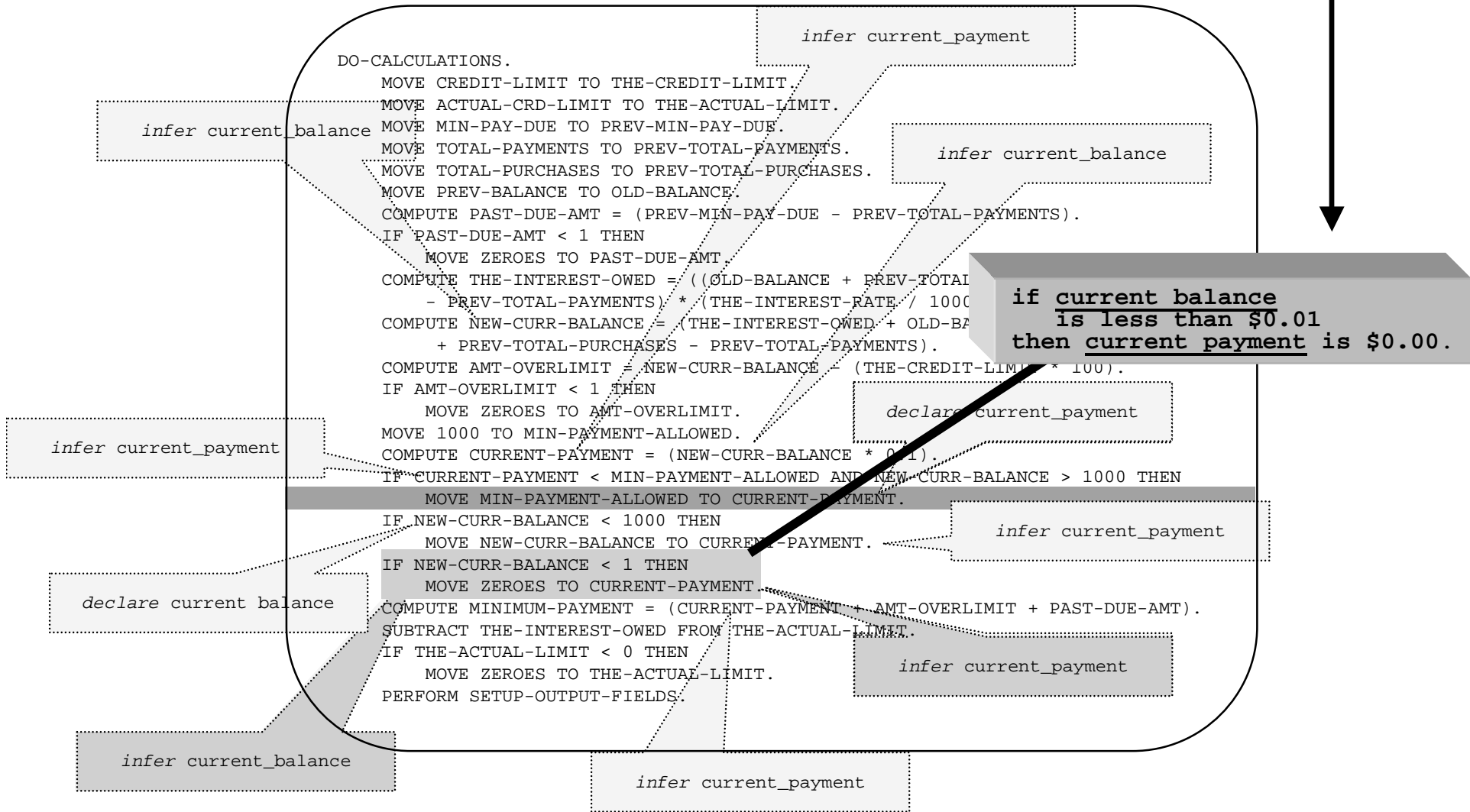
2) Interpret source code symbols



Source Browser and Annotation Tool

- What's the whole problem?
- What are business rules?
- Tools to extract rules

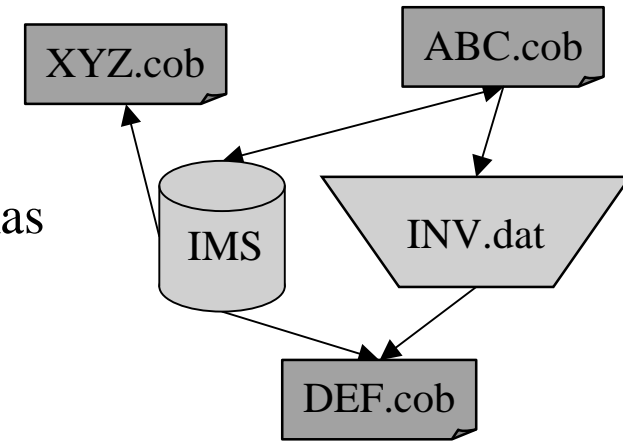
3) Construct business rule



- What's the whole problem?
- What are business rules?
- Tools to extract rules

Entity Relationship (ER) Extraction

- Application data model hiding in:
 - Hierarchical and relational database schemas
 - COBOL CopyLibs
 - Flat files between applications
- Automatic extraction of raw data model:
 - Hierarchical schema, integrated with Copylib declarations
 - Program Schema
 - Flat file data models
 - Identified by COBOL program read/writes and Job Control sequencing
- Directed transformation of raw data model into ER
 - Ties raw data model elements to business concepts
 - Ties indexes to essential business facts

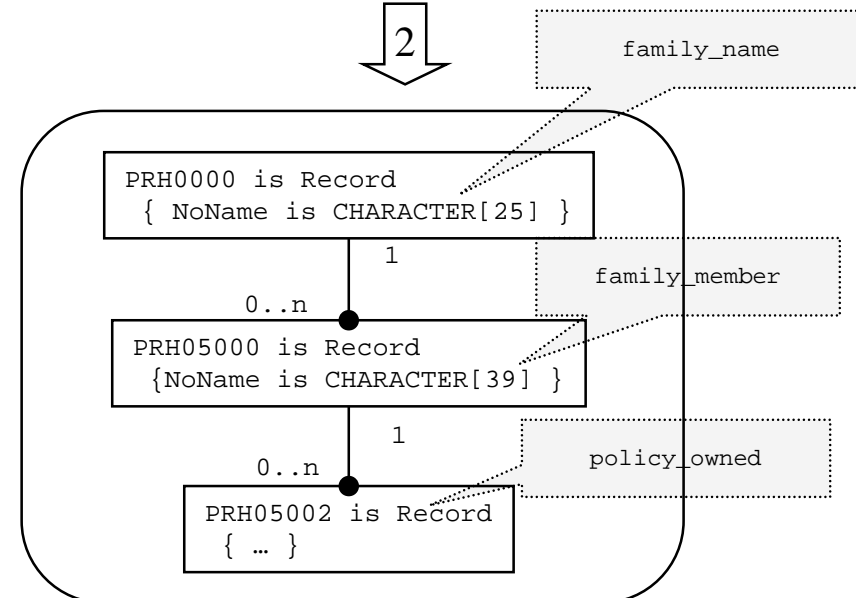
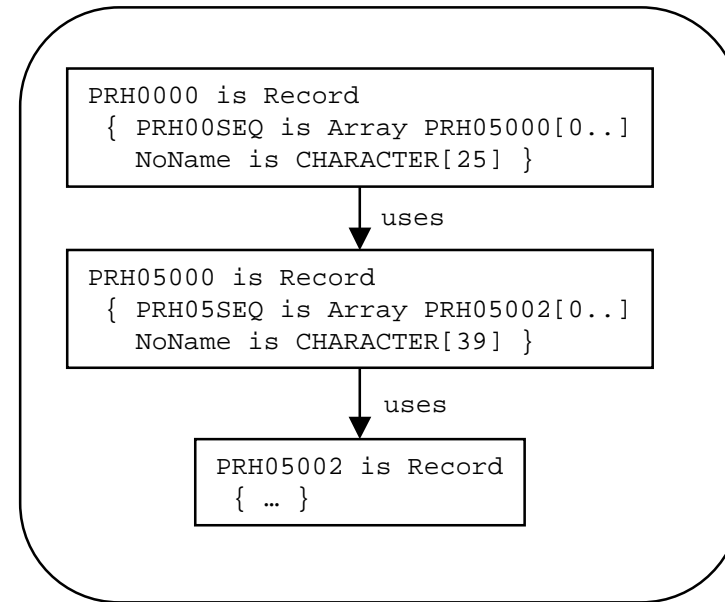


ER Extraction by Automated Transforms

```
// IBM IMS Examples
SEGM NAME=PRH00000, ←
  PARENT=0,
  BYTES=43,
  RULES=(LLL, LAST),
  PTR=(TWIN, , , ,),
  COMPRTN=(DPIEXIT, DATA, INIT)
FIELD NAME=(PRH00SEQ, SEQ, U),
START=1,
BYTES=18,
TYPE=C

SEGM NAME=PRH05000,
  PARENT=((PRH00000, SNGL)),
  BYTES=45,
  RULES=(LLL, LAST),
  PTR=(TWIN, , , ,),
  COMPRTN=(DPIEXIT, DATA, INIT)
FIELD NAME=(PRH05SEQ, SEQ, U),
START=1,
BYTES=6,
TYPE=C

SEGM NAME=PRH05002,
  PARENT=((PRH0500, SNGL)),
  ...
```



Summary

Business Rule Extraction using tools

- Tools for Business Analyst to understand code
 - Test Coverage
 - System Browsers, System Wide Information Flows
 - Entity Relationship Extraction
- Combine with Business Rule Assistant tool for Business Analyst to express rules
 - Annotation of code with concepts and BR fragments
 - Capture of Business Rules in SBVR, an OMG Standard
 - Checking of rules for errors
- *Technology is ready to do all this*
- Benefits
 - Extraction of business vocabulary and rules from code
 - Traceability of business rules back to code
 - Enables rationalization of business rules with organization
 - Traceability enables change of code enforcing business rules
 - Synergy with downstream reengineering