
# A Business Rules-Based Approach to Information Systems Development

Dr. Paul Dorsey

Dulcian, Inc.

www.dulcian.com

NYOUG – December 12, 2002


# Agenda

- Part I: Overview & Background of Business Rules
  - Definitions
  - History
  - Advantages/Disadvantages of approach
- Part II: Quest to Create a Rules-Based Development Environment
  - Implementing business-rule based systems


# It's all about business rules

- In the last 5 years, business rules have emerged at the center of the application development process.
- There are now entire conferences and publications dedicated to business rules:
  - Business Rules Forum 

  - ODTUG Business Rules Symposium 

  - BR Community 

- Lots of business rules papers at every conference.

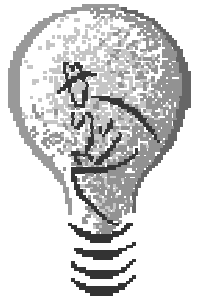

# Business Rules

- New, cool buzzword?
  - Same old stuff in a new package?
- Hype?
  - Remember CASE, artificial intelligence…etc.
- Real paradigm shift?
  - Completely revolutionize database industry

Answer: Maybe all three

# What are business rules?

- A new word for requirements ??
  - Almost…
  - Refocus from “requirements” to “rules”
- Figuring out the business rules is what what good analysts have always done.
- Business Rules include *all* business requirements



A precisely articulated requirement
that is both machine readable & user readable.

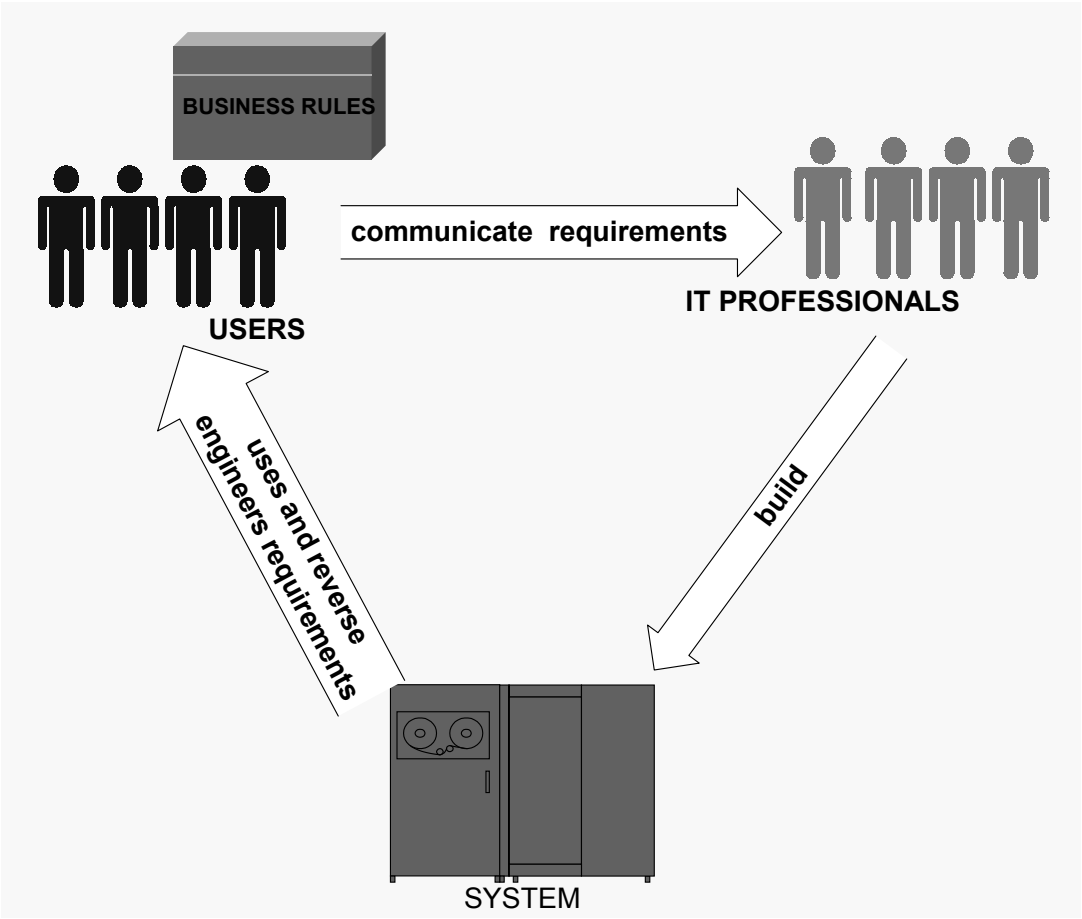
# History of the Leading Designers (1)

- 1980-1990 Denormalization
  - "There is no way to get good performance out of a relational database that is normalized. Good, skilled denormalization is the key."
- 1990-1995 Normalization is OK
  - "Now that the database is fast enough, we can normalize to make our systems easier to build."

# History of the Leading Designers (2)

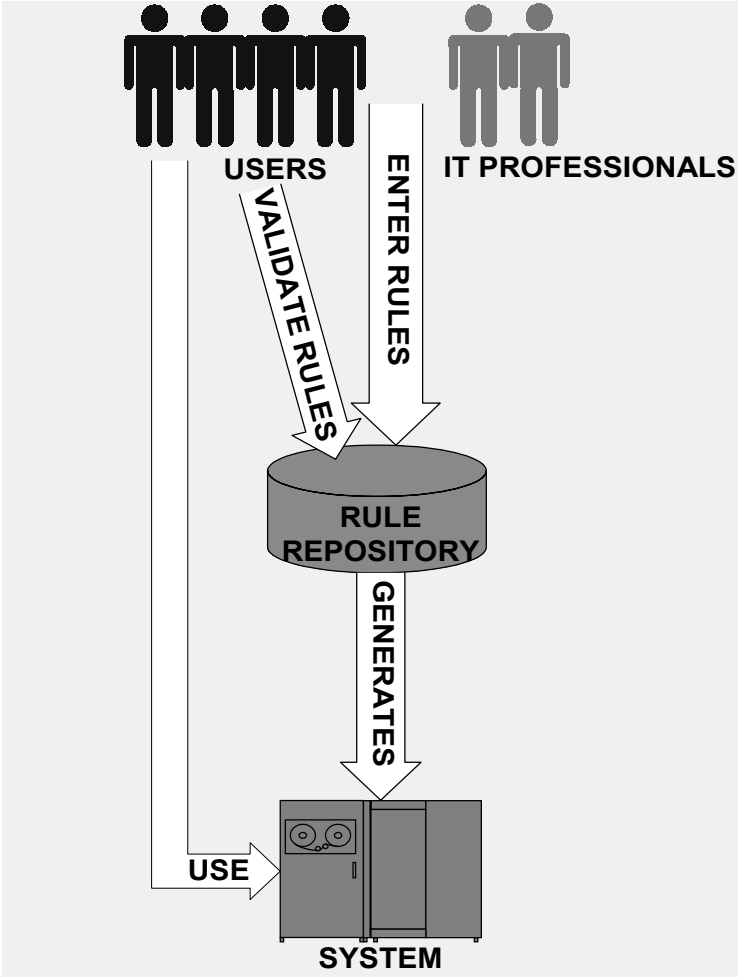
- 1995-2000 Generic Data Models
  - "We can decrease the size of our databases by modeling some quantity of our systems as data use rule tables."
- 2001-Present Business Rules-Based Systems
  - "Put as many rules as possible into a repository and generate as much of the system as possible."

# Flawed Communication in the System Design Process




There is no common point of communication between Users and IT Professionals.

# Communication using a business rule approach




The Rule Repository is the common point of communication between Users and IT Professionals.


# The REAL Advantage!!!!

- “The only reason you are able to build so cheaply is that you foist the programming off onto your users.”

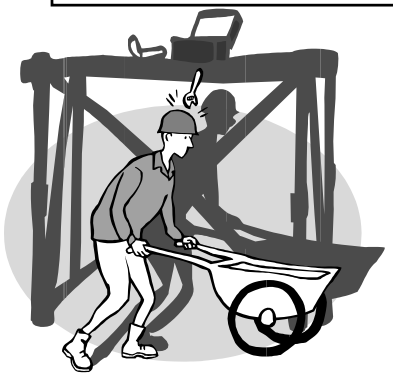
  Ulka Rodgers

# Building the System

## Traditional

- Strategy
- Analysis
- Design
- Build
- Test

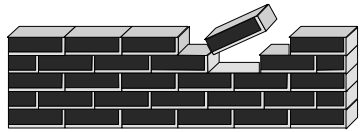
Multi-week cycle time



## Rules-Based

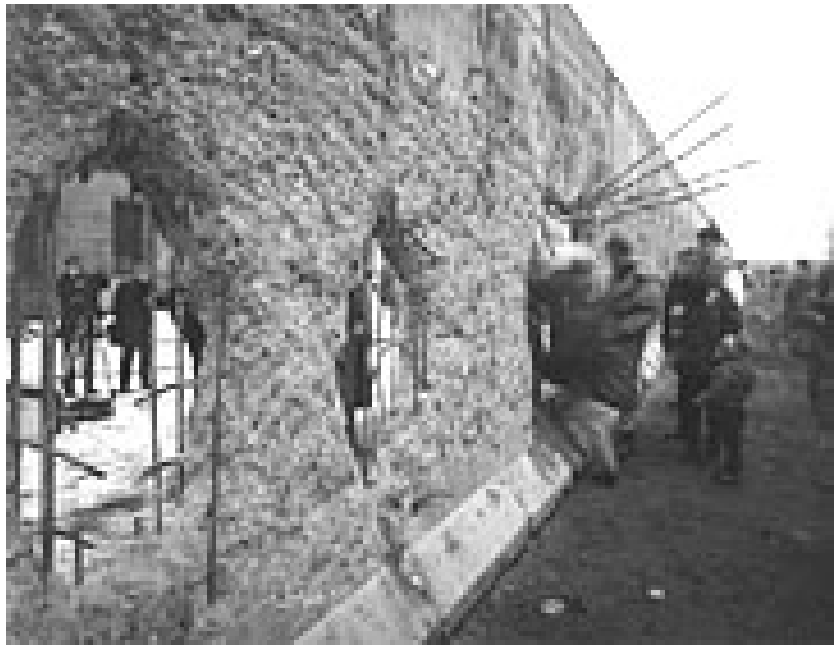
- Strategy -- no change
- Analysis (Rule Gathering and validation)
  - Gather rules
  - Implement system
  - Test
- Build Custom Applications

Zero cycle-time



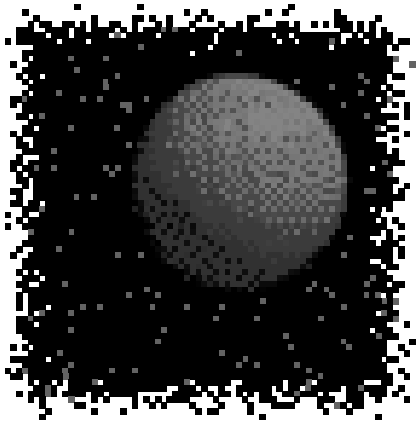
# Business Rules approach is not a new idea

- Existing business rule-based approaches:
  - 1. Reference tables
  - 2. Data modeling tools
  - 3. Generic modeling
  - 4. Existing business rule tools
- All are fragmented approaches.




# Imagine Planet Business Rule where…

- Users are enabled to articulate their requirements (aka business rules).
- We can easily gather and place requirements into a business rules repository.
- The whole system is generated from the business rule repository.



# Is Planet Business Rules a trap?

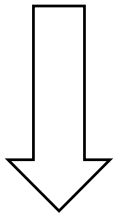
- Rules must be manageable!
  - 500,000 sentences is not viable.
  - Flowcharts with 3000 boxes don't help.

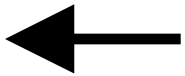
# How can you manage business rules?

- 1. Analysis Rules ≠ Implementation Rules
  - Analysis rules = how users talk
  - Implementation Rules = precise specification
- 2. Rules must be partitioned.

# Business rule-based systems development

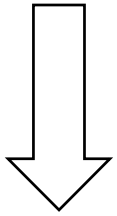
- Analysis Rules
  - Declarative
  - Business ramblings





Hard, creative transition

- Implementation Rules
  - Structural (Data Model)
  - Process (Process Flows)



Trivial transition

- Final system

# Hard Questions to Ask

- 1.Where will all of the rules go?
  - Database triggers, procedures
  - Rules Product
  - Applications
- 2. How will the system interact with the DBMS?
  - Extracting to rules tool is terrible.

- 3. How will very detailed security be handled?
  - Who can do what in each state?
- 4. What about configuration management?
  - Real time update is not enough.
- 5. How can the system be integrated with existing applications?
  - Integration must be seamless.


# Tool Types

## Analysis Rule Gathering

- Capture rule paragraphs
- Sort into categories
- Generate standard Analysis Document

## Implementation Rules

### Process Flows

- Flowcharts, STE
- Look great in demos
  - Need hundreds of boxes for production flow
- Don't do as much as they seem
- Generate code
- Usually are not integrated into DBMS

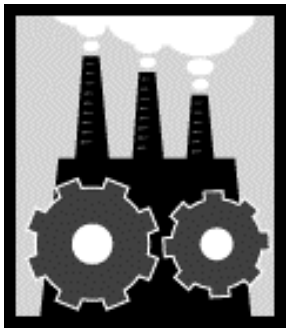
### Database Triggers

- Specify structural rules
  - "Start date < End date"
- Generate trigger code
- Limited use

# The Industry

- Analysis rule gathering
  - Business Rules Solutions
    - Framework Solutions
  - BRIM™
    - Dulcian, Inc.
- Process Flow (Biggest group)
  - ILOG
  - Savant
  - Seeristic
  - BRIM™

- Database trigger
  - Nintech
  - Seeristic
- Full system generation
  - BRIM™




# Need for Technical Leadership

- Every successful business rules-based implementation has used a senior architect.
- Why does business rule-based development require a good tech lead?
  - 1. Requires getting pieces to work well together (although there are fewer pieces)
  - 2. Totally model-driven design
  - 3. True paradigm shift even if you are using traditional tools
- The moral of the story is:
  - You just can't buy a tool and run with it.

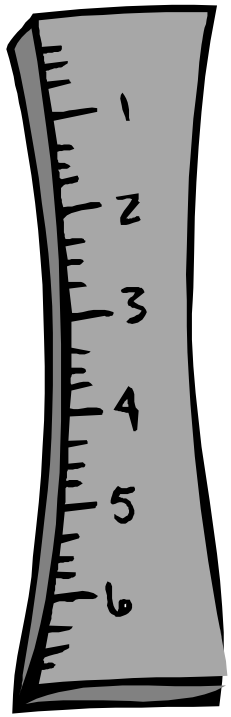
# How to guarantee BR project failure

- 1. Attempt to bolt in a tool
  - Don't think through the whole process
- 2. Inadequate tech lead
- 3. Insist that "We don't need any help."
- 4. Get rid of tech lead too soon.

- 5. Start writing lots of code to overcome deficiencies in rule representation.
- 6. Use declarative process rules
  - Process rules require a process flow.
- 7. Don't worry about configuration management
  - Cost of V2 is huge
  - Full data migration

# The Business Rules Community

- 2 main groups
  - The originals:
    - Ron Ross, Terry Moriarty
    - Business Rules Forum
    - Focus on Analysis rules, grammars, taxonomies
  - The builders:
    - Paul Dorsey, Roland Berg, Dave Wendelken
    - ODTUG Business Rules Symposium
    - Focus on building systems
    - Tools, utilities, code/DB generators
- Can't we all just get along?



# Case Study
# US Air Force Recruiting – V1

- Air Force Recruiting
  - Tools/Approach
    - Oracle Designer
    - Generic Modeling
    - Declarative Rules
  - Cost $5million range
  - Results
    - Success
    - When tech leads left – project struggled
  - Lessons learned
    - It's hard to do on your own
    - Full rule support is very hard on a home-grown system.

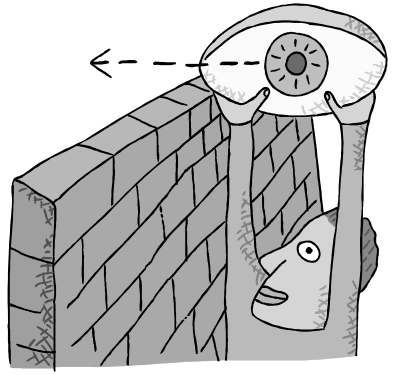

# Case Study
# US Air Force Reserves
# Recruiting – V1

- Air Force Recruiting
  - Tools
    - BRIM™
    - PureEdge (front-end tool)
  - Cost $1million range
  - Results
    - Looking good so far
    - Things are going very fast.
  - Lessons learned
    - Analysis rules are VERY important.
    - Map analysis rules to implementation rules
    - Having all rules in a single repository makes them independent of the physical implementation.


# Part II:
# The Quest to Create a Complete Business Rules-Based Development Environment

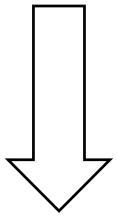

# The Vision

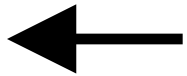
- All rules go into a repository.
- Rules grammar is robust enough to specify the whole system.
- Users can read (and help enter) rules.
- You can generate the whole system.
  - The rules ARE the system.
- The generated system will be almost the same as the system you would build by hand.



# Recall: Business rule-based systems development

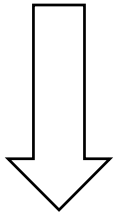
- Analysis Rules
  - Declarative
  - Business ramblings





Hard, creative transition

- Implementation Rules
  - Structural (Data Model)
  - Process (Process Flows)



Trivial transition

- Final system

# Analysis Rules

- Everyone thinks about rules differently.
- Users think in text.
- Let each project team design its own analysis rules structure.
- Map analysis rules to implementation rules.


# Implementation Rules

Partition the rules:

| Structural Rules: Data Model | Process Rules: Process Flows |
|---|---|

- This is the way we have been thinking for years.
- We just need to improve upon the idea.


# UML as a business rule language

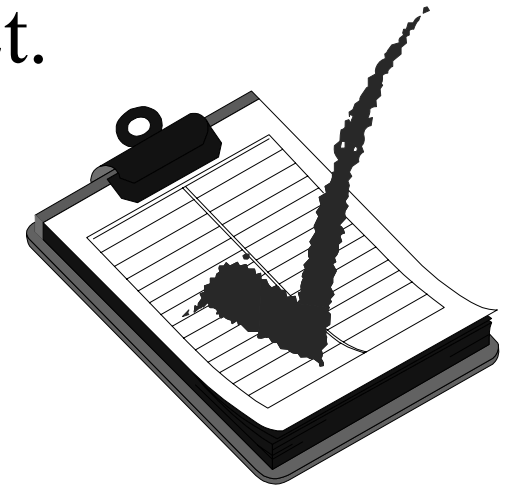
- Unified Modeling Language
  - Oracle's direction
  - Established standard
  - Extensible
  - Structural rules
    - Class diagrams
  - Process rules
    - Activity diagrams

# Structural Rules – the problem

- ERDs – Can be improved upon
  - Not enough rules included
  - Not extendable
  - Reference tables aren't nice
  - No inheritance

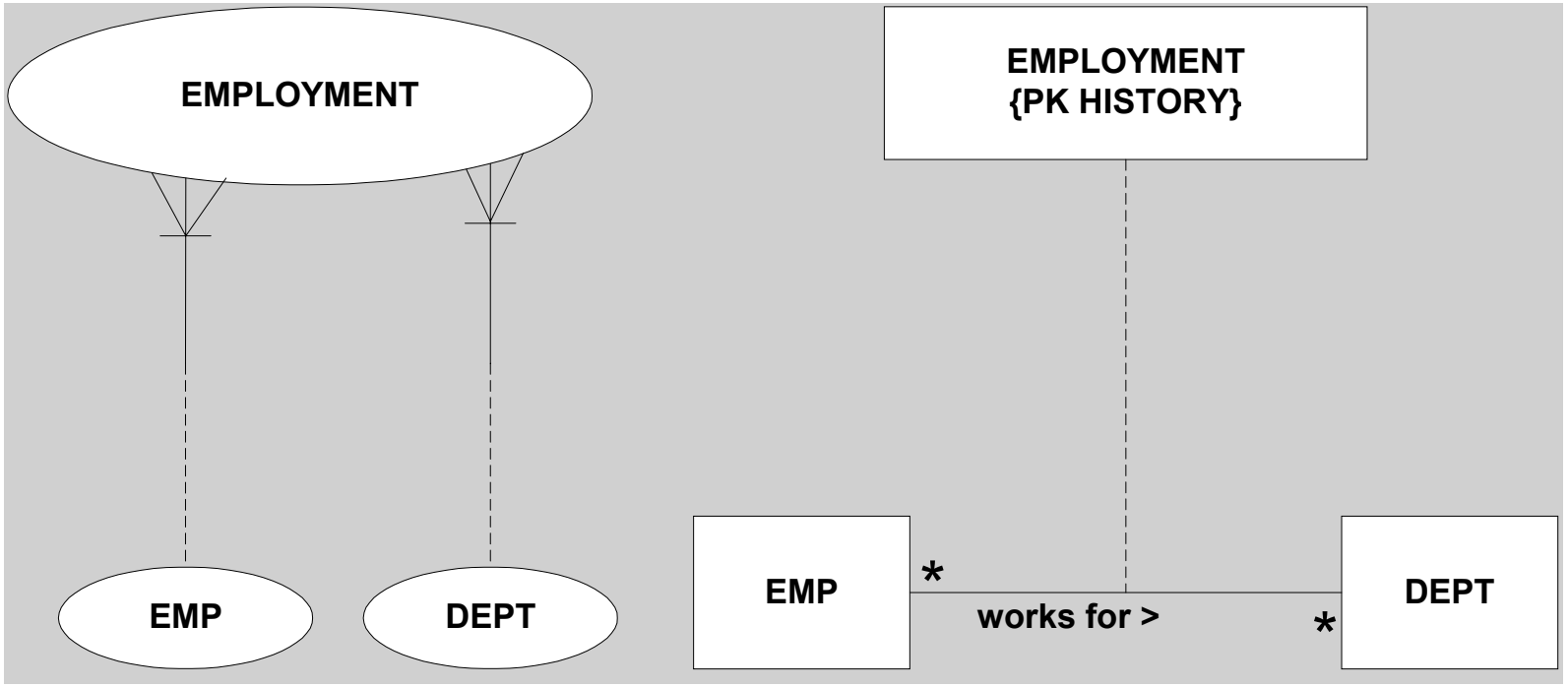
# Structural Rules – the solution

- Use UML Class diagrams with extensions:
  - Standard
  - Extensible
  - Superset of ERDs (Easy to move to)
- Makes rule representation compact.



# A Better Way to Model

- 1. No reference tables
- 2. More precise and complete



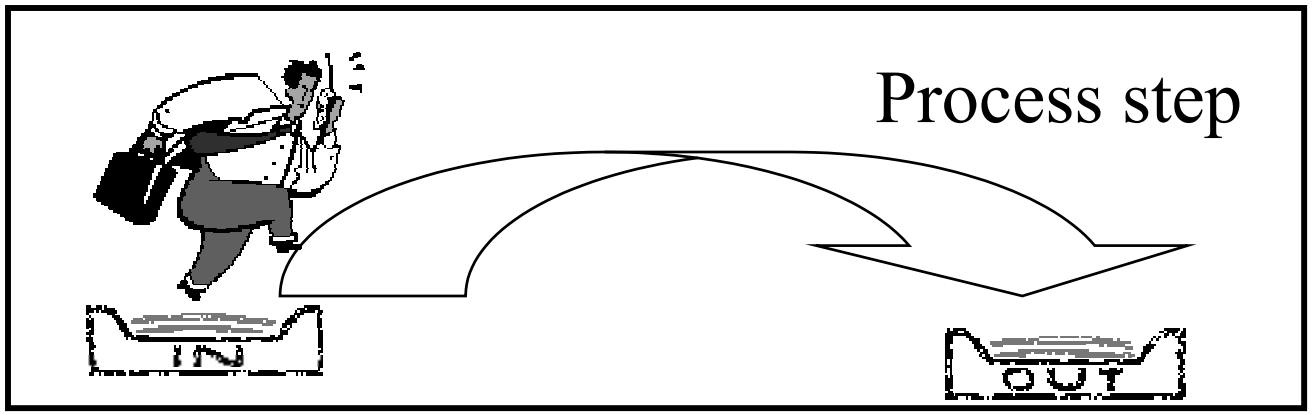
# Process Rules – the problem

- Declarative rules are hard to implement:
  - "Purchase orders over $50,000 require three levels of approval."
  - Too many rules
  - Rules interact
  - Terrible performance
  - Example:
    - 14,000 rules
    - No idea of process
    - No way to debug
    - Poor performance

- Traditional flowcharts or state diagrams are also difficult:
  - Too many little boxes
  - Decomposition only moves the boxes around.

# Process Rules – the solution

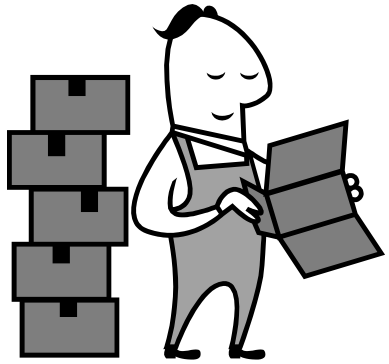
- Business-based process flows
- Try a user's idea of a "state."



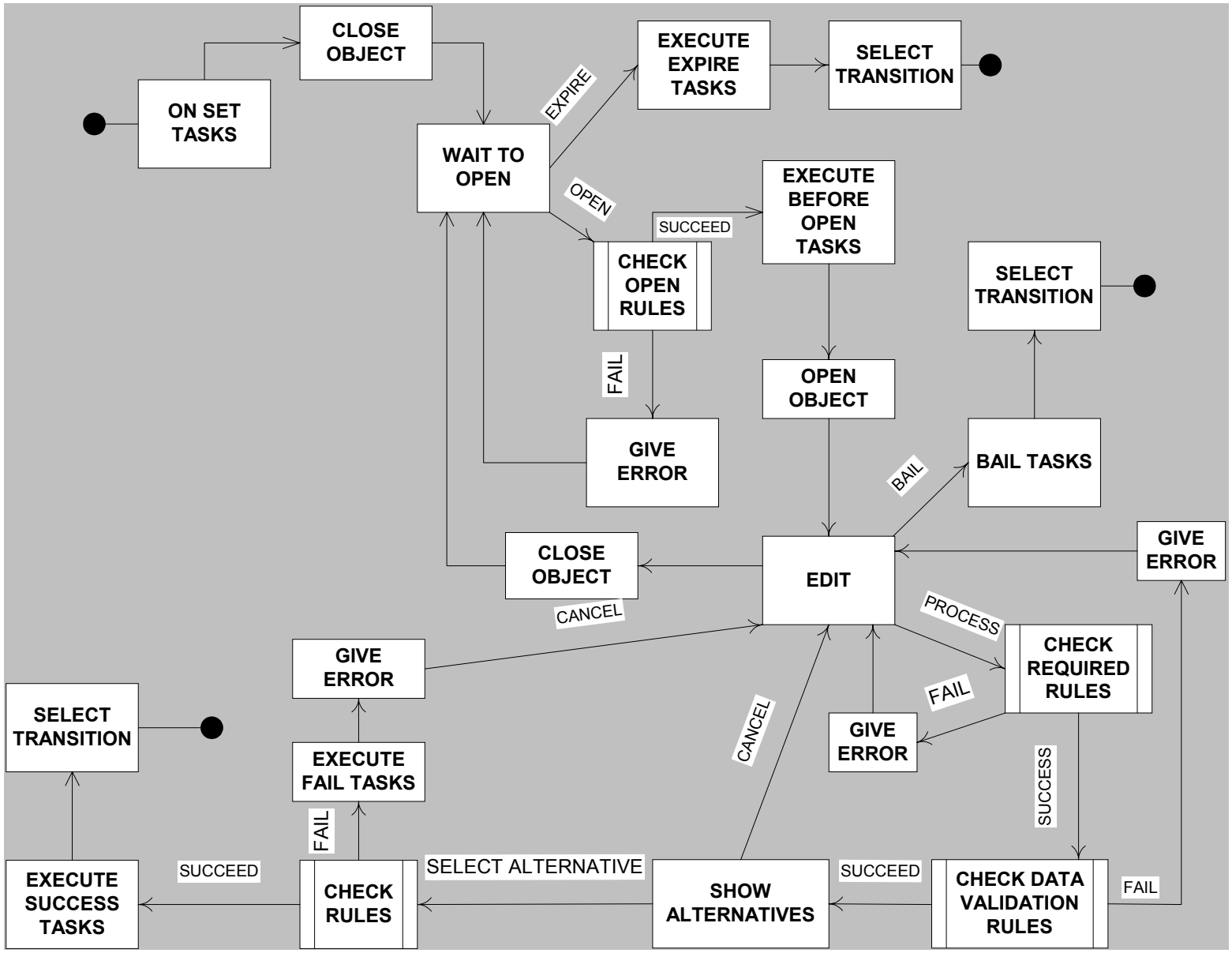

- One user state replaces 20-30 state transition engine states.
- Makes rule representation compact.


# States Belong to Objects

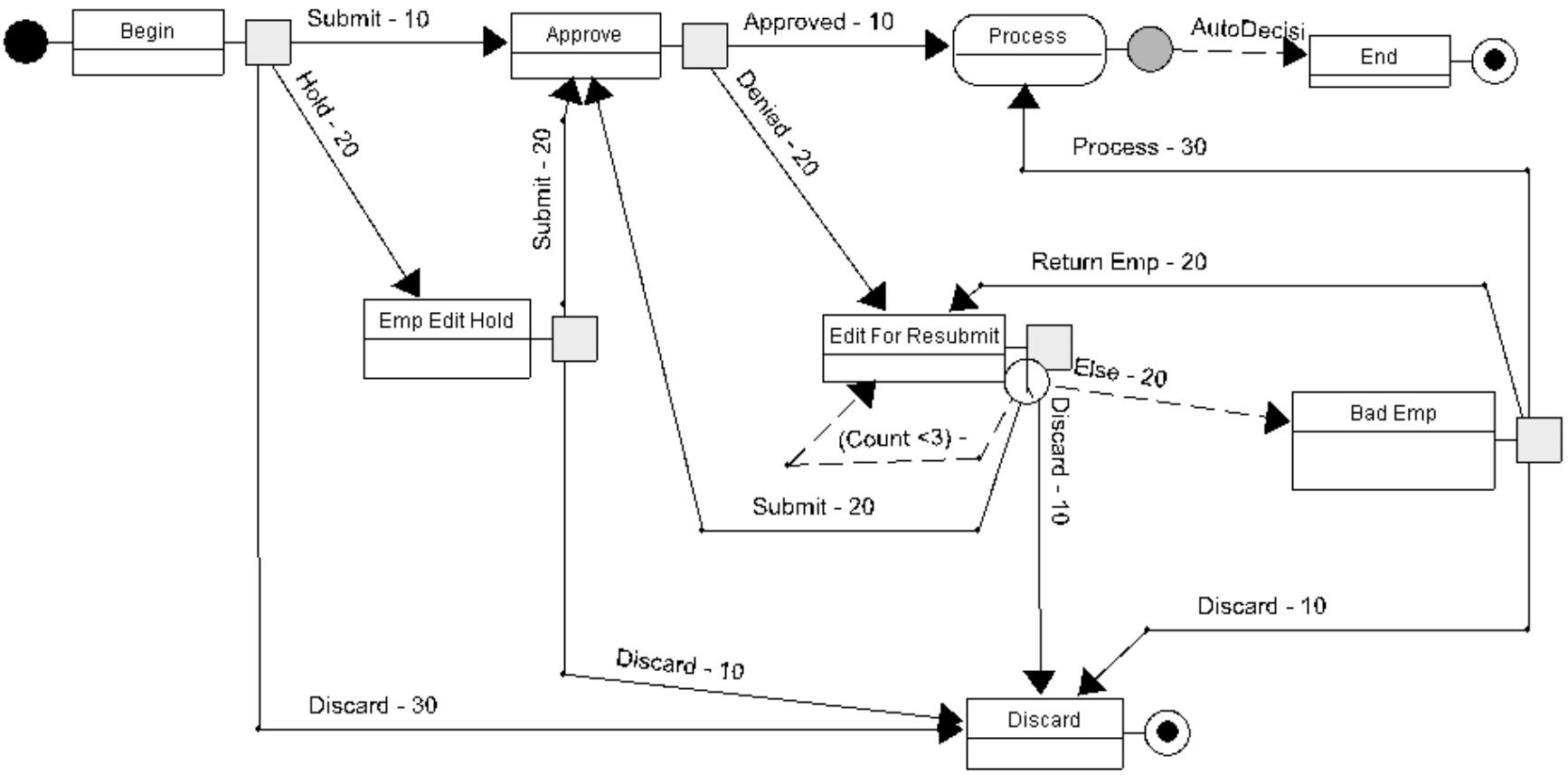
- Application-independent
- No rules in applications
- Fully integrated with data model
- “Classes have attributes. Classes have security rules. Some classes have states.”



# “Inbox State” flowchart

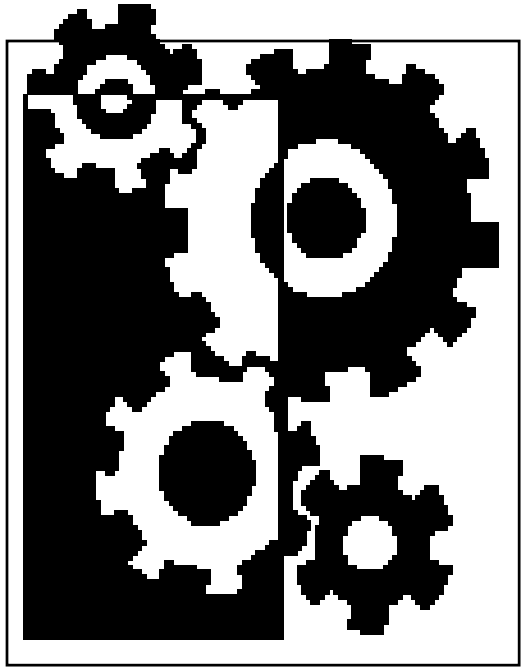


# Timesheet Flow



# System Generation: Structural Rules

- Classes are tables.
- Attributes are columns.
- Reference tables are domains.
- Should be as close to traditional DBMS as possible
- Generate:
  - tables
  - views
  - triggers




# System Generation: Process Flows

- Generate procedures to run in:
  - Database
  - Middle tier
  - Client
- Code must be generated to provide adequate performance.
- Application independent
- Application should call “ProcessObject( )”

# System Generation: Making it work

- Planet Business Rule is just a playground without:
  - Configuration management
  - Rich security model
  - Full integration with DBMS


# Conclusions

- Business rule-based design and development is the next big shift in the industry.
- Using an existing commercial product is, at best, a fragmented approach.
- Complete rules-based environments are coming.
- The key to project success is a good technical lead.


# Dulcian's BRIM Environment

- Full BR based development environment
- Freeware
- For Demo/License
  - Write "BRIM" on business card


# Finding out more about Business Rules-Based Systems

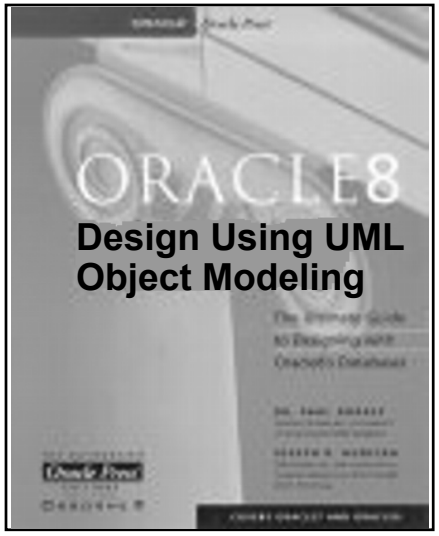
- Join the Business Rules Advisory Group
  - Free membership
  - Discounts on BR products, conferences
  - Free listserv run by ODTUG
  - Write "BRAG" on business card
- Business Rule Conferences
  - Business Rules Forum
    - November 4-8, 2002 – New Orleans, LA
  - ODTUG Business Rules Symposium
    - June 15, 2003 – Miami Beach, FL
- BRIM™ is shareware
  - Complete business rules-based environment
  - Free, unrestricted site licenses
  - For free demo, write "BRIM" on business card




# Contact Information

- Dr. Paul Dorsey – paul_dorsey@dulcian.com
- Dulcian website - www.dulcian.com







Coming by year end
Oracle 9i JDeveloper Handbook