Constraint Definition Language in Oracle Configurator – A Holistic Perspective

Krishnan Sundaresan
Fujitsu Consulting, Inc.

Session Objectives

- To expound the features of Constraint Definition Language – its terminology, syntax, grammar and the like
- To contrast the CDL approach with the interactive rules definition for typical business scenarios
- To explicate the advantages, challenges and implementation considerations

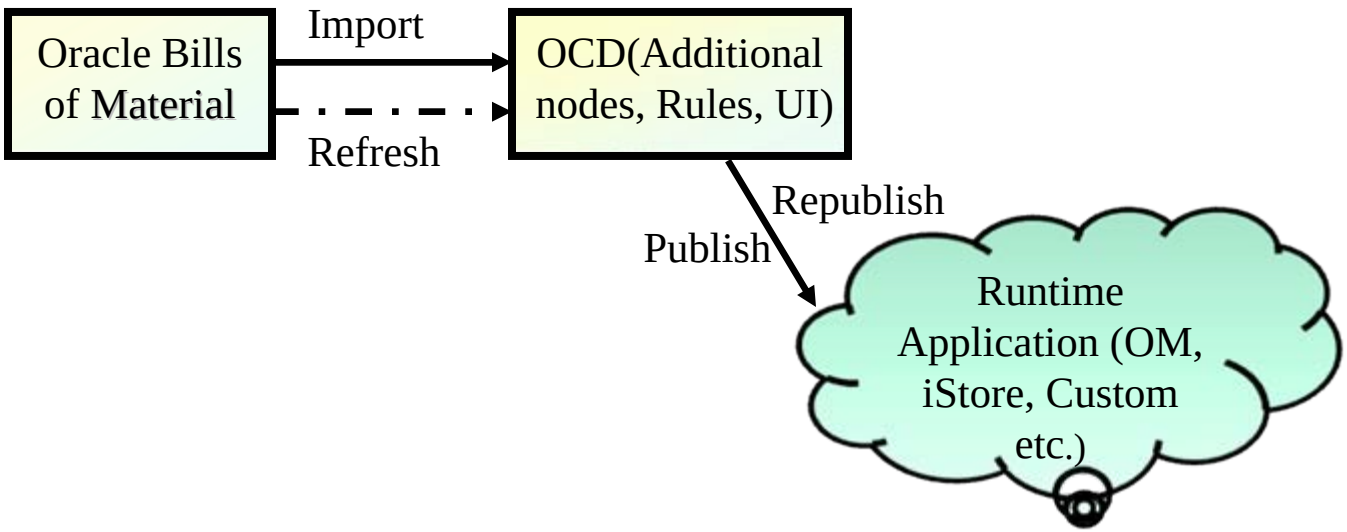
Oracle Configurator – An overview

- To create a valid product based on design and customer requirements
- Enforces rules and constraints resulting in a valid product
- Source is Oracle Bills of Material
- Assemble to Order, Pick to Order and Hybrid models

Oracle Configurator – An overview

- Define BOM with Option Classes and Options
- Import BOM into CZ
- Define rules and UI
- Test and Publish
- Calling Applications – Order Management, iStore, Quoting, Custom web application etc.
- Refresh, redefine and republish

Oracle Configurator – An overview



Configurator Rules

- Constraining conditions for a product
- Requires significant product expertise and knowledge of end customer requirements
- Finalized during product design and modified as required
- Various types of rules that are supported in Oracle

Configurator Rules

- Logical – Requires, Implies, Defaults, Defaults, Negates
- Numeric – quantitative relationships
- Comparison – compare numeric relationships
- Compatibility – decide validity of other items
- Configurator Extensions – custom logic for business-specific requirements

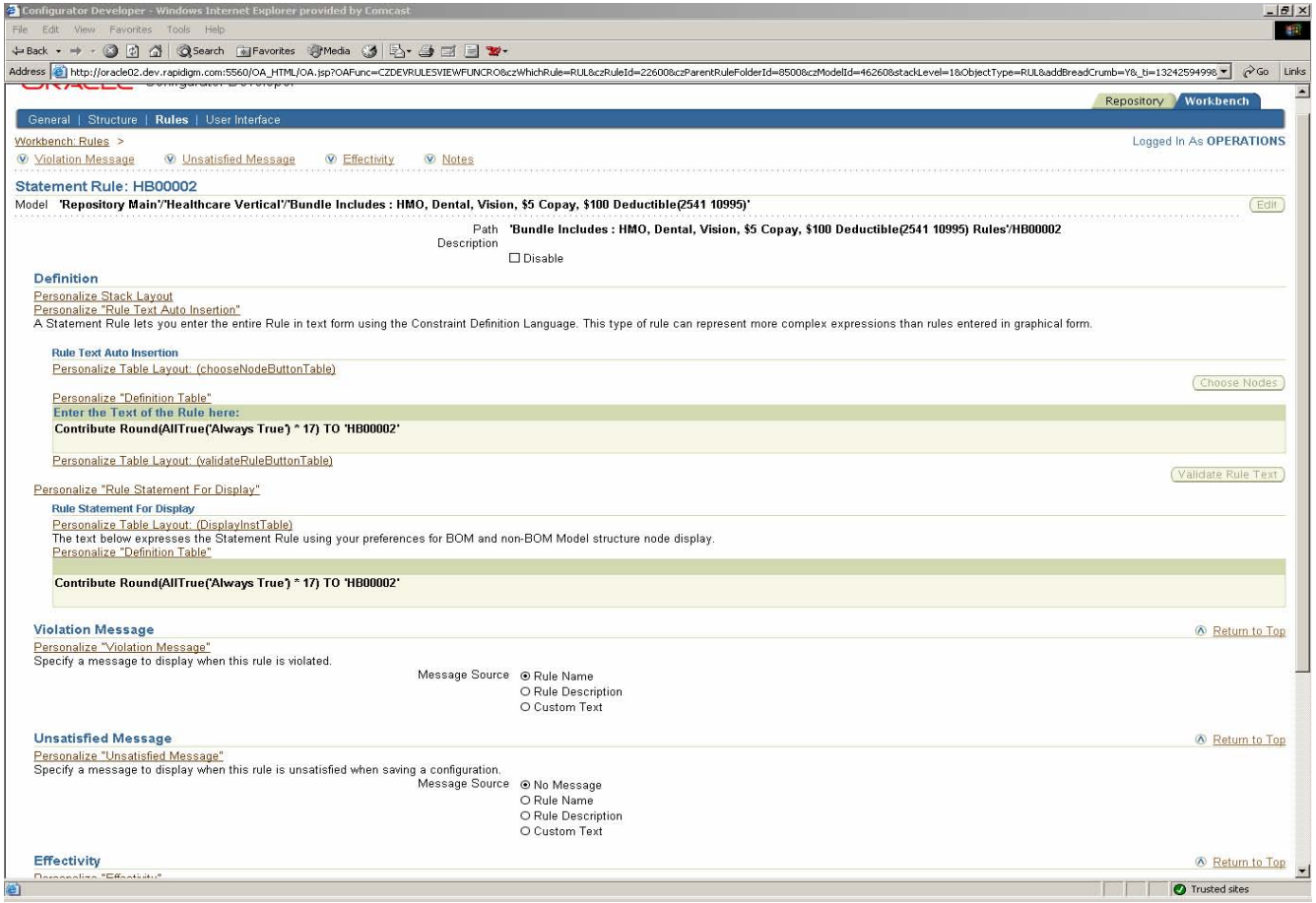
CDL – Building Blocks

- Purpose of rule, participants and expressions
- Flexibility and Versatility
- Model definition – Design/Manufacturing considerations
- Simplify BOM structure
- Statement rules in CDL – multiple rules and operands in a single rule

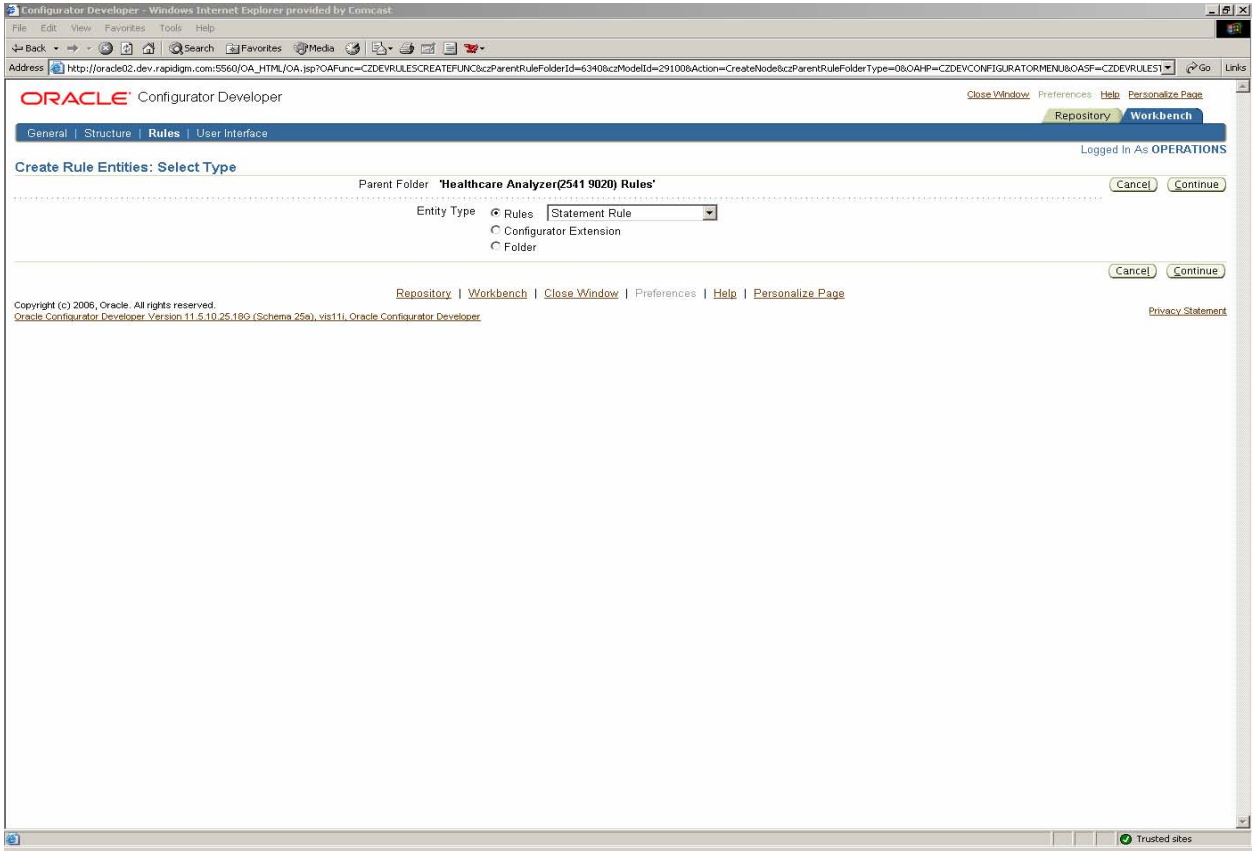
CDL – Building Blocks

- Elements – rule, statements, case sensitivity, quotation marks, comments
- Definition – Name, Description, Effectivity, Usage
- Statement – rule content
- Case sensitivity – nodes
- Quotation marks – when operand same as node

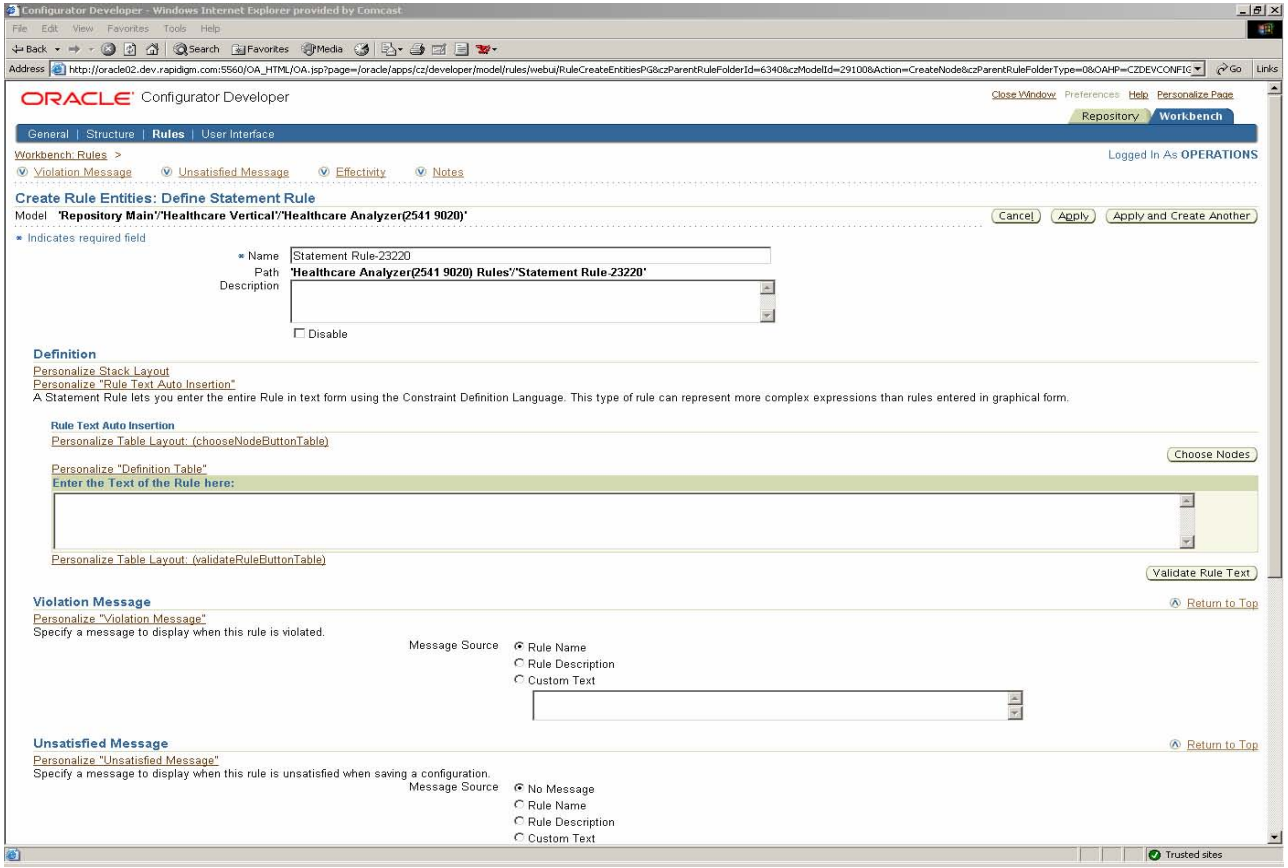
CDL – Building Blocks



CDL – Rule Definition



CDL – Rule Definition



Explicit and Iterator statements

- Explicit – participants identified explicitly
- Rule confined to these participants
- Iterator – range of nodes based on properties
- FOR ALL IN, WHERE
- Rule applies to every node that satisfy properties
- Complement Compatibility rules

Expressions

- An operator and two operands
- Advanced Expressions in previous releases
- X AND (Y OR Z) – two expressions
- Mathematical expressions for calculation, comparison
- Efficient and accurate structuring

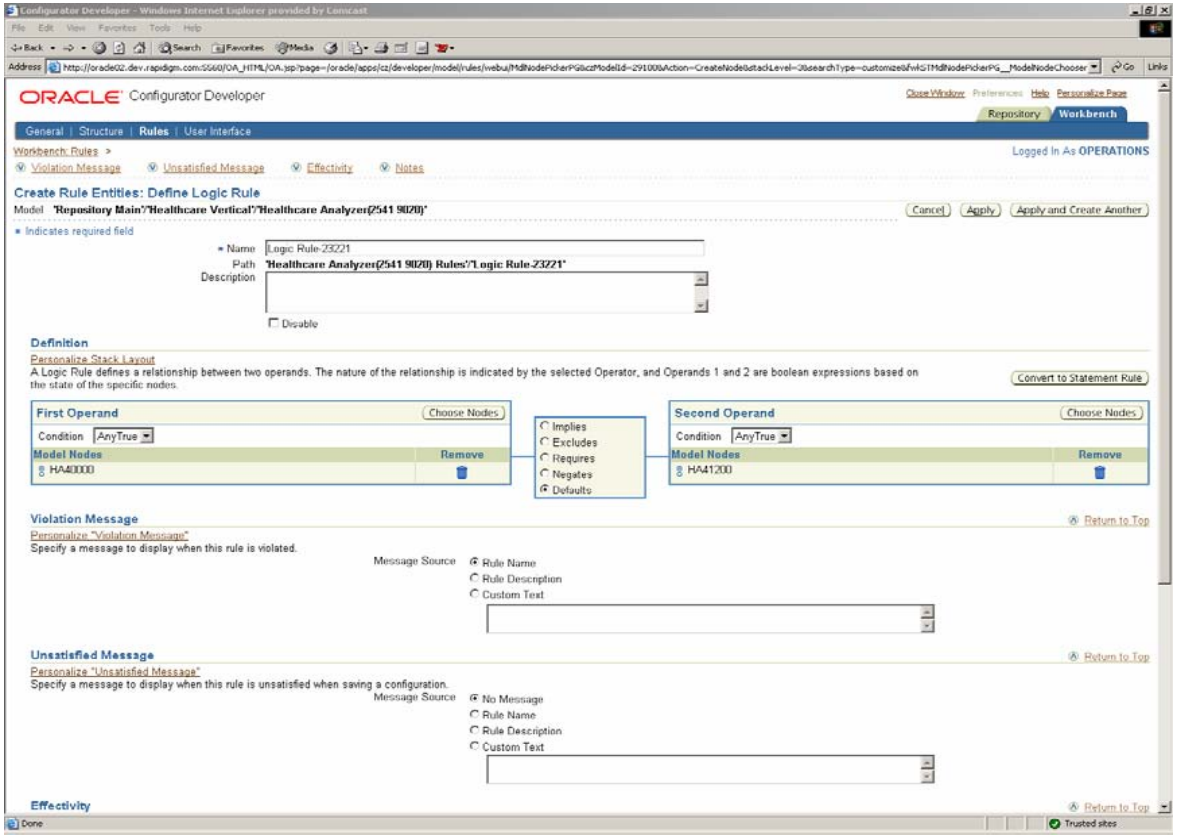
CDL Keywords

- CONSTRAIN – optional in certain conditions
- CONSTRAIN X IMPLIES Y
- CONTIRBUTE…TO
- CONTIBUTE X TO Y
- COMPATIBLE
- COMPATIBLE X OF Y

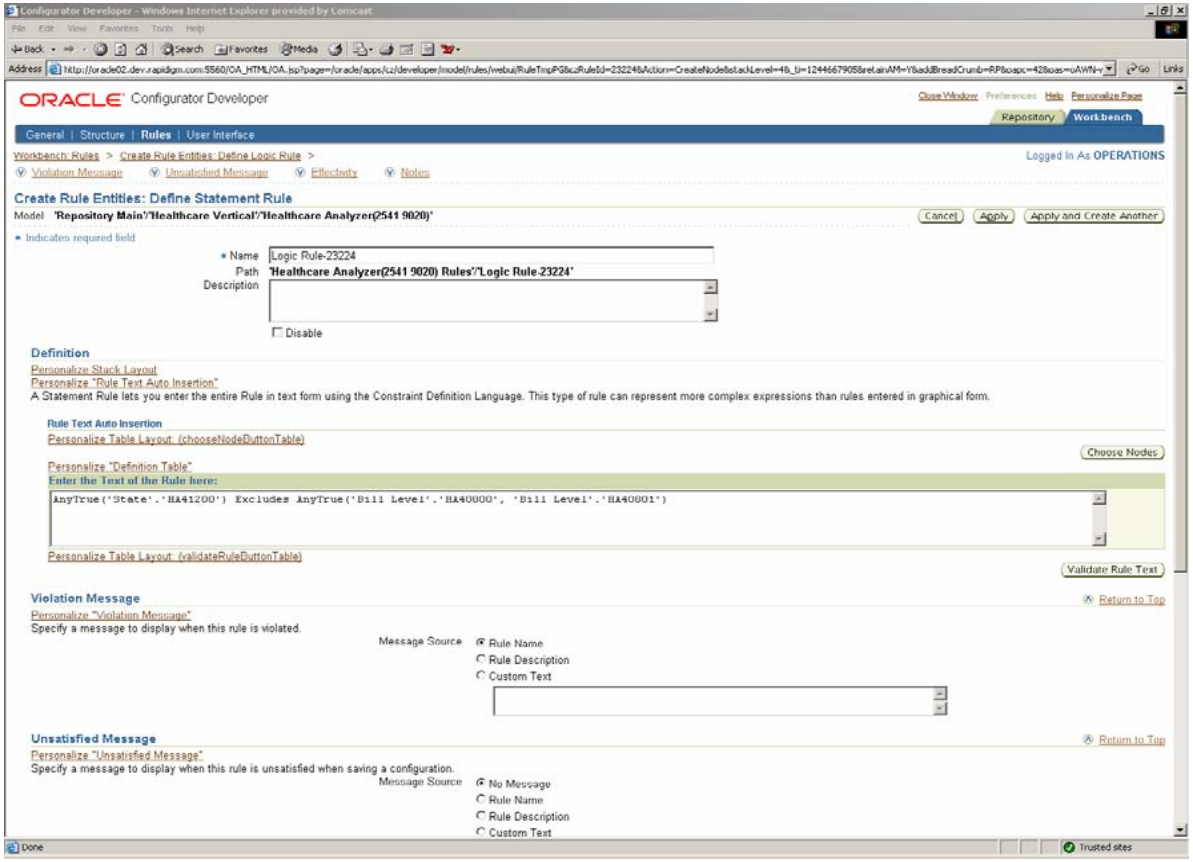
CDL and Non-CDL Rules

- Advanced Expressions necessitate CDL
- Simple expressions – non-CDL
- Comparison
- X DEFAULTS Y
- X EXCLUDES Y AND Z

Non-CDL Rule


ORACLE Configurator Developer
Repository
Workbench
General | Structure | Rules | User Interface
Violation Message
Unsatisfied Message
Effectivity
Notes
Create Rule Entities: Define Logic Rule
Cancel
Apply
Apply and Create Another
Name
Logic Rule-23221
Description
Disable
Definition
Personalize Stack Layout
A Logic Rule defines a relationship between two operands. The nature of the relationship is indicated by the selected Operator, and Operands 1 and 2 are boolean expressions based on the state of the specific nodes.
Convert to Statement Rule
First Operand
Choose Nodes
Condition
AnyTrue
Model Nodes
Remove
HA40000
Implies
Excludes
Requires
Negates
Defaults
Second Operand
HA41200
Violation Message
Return to Top
Personalize "Violation Message"
Specify a message to display when this rule is violated.
Message Source
Rule Name
Rule Description
Custom Text
Unsatisfied Message
Personalize "Unsatisfied Message"
Specify a message to display when this rule is unsatisfied when saving a configuration.
No Message
Effectivity

CDL Rule


Configurator Developer - Windows Internet Explorer provided by Comcast
ORACLE Configurator Developer
Close Window Preferences Help Personalize Page
Repository Workbench
General | Structure | Rules | User Interface
Workbench: Rules > Create Rule Entities: Define Logic Rule >
Logged In As OPERATIONS
Violation Message Unsatisfied Message Effectivity Notes
Create Rule Entities: Define Statement Rule
Model 'Repository Main'/'Healthcare Vertical'/'Healthcare Analyzer(2541 9020)'
Cancel Apply Apply and Create Another
Indicates required field
Name Logic Rule-23224
Path 'Healthcare Analyzer(2541 9020) Rules'/'Logic Rule-23224'
Description
Disable
Definition
Personalize Stack Layout
Personalize "Rule Text Auto Insertion"
A Statement Rule lets you enter the entire Rule in text form using the Constraint Definition Language. This type of rule can represent more complex expressions than rules entered in graphical form.
Rule Text Auto Insertion
Personalize Table Layout: (chooseNodeButtonTable)
Choose Nodes
Personalize "Definition Table"
Enter the Text of the Rule here:
AnyTrue('State'.'HA41200') Excludes AnyTrue('Bill Level'.'HA40000', 'Bill Level'.'HA40001')
Personalize Table Layout: (validateRuleButtonTable)
Validate Rule Text
Violation Message
Return to Top
Personalize "Violation Message"
Specify a message to display when this rule is violated.
Message Source
Rule Name
Rule Description
Custom Text
Unsatisfied Message
Return to Top
Personalize "Unsatisfied Message"
Specify a message to display when this rule is unsatisfied when saving a configuration.
Message Source
No Message
Rule Name
Rule Description
Custom Text
Done
Trusted sites

Implementation Considerations

- Product structure, complexity
- Boolean operator rules – CDL mandatory
- During upgrade to 11i10 and beyond
  - Advanced Expressions converted
  - Functional Companions to Configurator Extensions
  - Logic regenerated for current rules
  - CONSUMES FROM modified
  - AnyTrue, AllTrue for Numeric rules - converted

Implementation Considerations

- Number of models, type and complexity
- Calling applications impact
- Order Management – setups and requirements
- Custom Application – totally different setups

Integration Touch Points

- Starting point in the business flow
- Manufacturing, Procurement, Inventory control, Planning etc.
- ATP and Pricing requirements
- Inventory controls
- Sourced from vendor, subsidiary
- Service Contracts
- Customer facing documents

Questions and Answers

Krishnan Sundaresan
Principal Consultant, Oracle Practice
Krishnan.Sundaresan@us.fujitsu.com