



Changing the rules of business™

**SEAMS Workshop
May 2006, Shanghai**

**Optimizing a Rule Engine
using Adaptive Programming Techniques**

**Changhai Ke
Chief Architect, ILOG
<http://www.ilog.com>**

Agenda



Changing the rules of business™

- Rule engines, rule interpreters
- Implementation & challenges
- 1. Dynamic rule compilation
- 2. Information provided by the users
- 3. Dynamically generated stand-alone rule engines
- Conclusions

Business Rules



Changing the rules of business™

What are they?

1. If the shopping cart contains 3 CDs and the total value is greater than \$100, then offer a free CD.
2. If the call is made during the first week of the chinese new year, give a 20% discount.

- Languages to express them
- Tools (graphic) to edit them
- Rule engines to execute them

An Example of Rule Editor





Changing the rules of business™

Business Rule: checkIncome

General Information

Name : checkIncome

Category Filter

 Categories: any.  [Edit](#)

Documentation

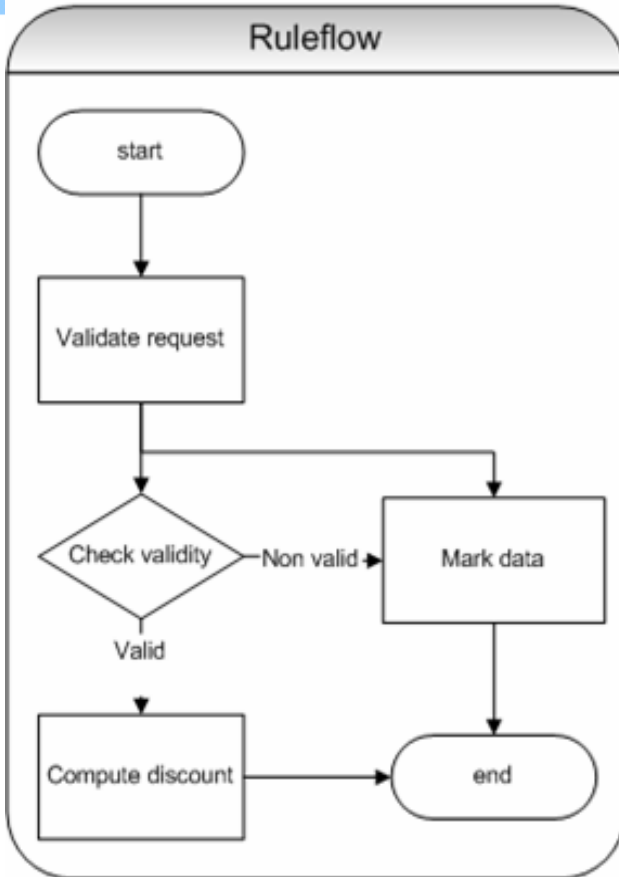
Code

```
definitions
  set 'minimum income' to 0.37 * the yearly income of 'the borrower';
if
  the yearly repayment of 'the loan' is at least 'minimum income'
then
  in 'the loan report', refuse the loan with the message "Too big Debt/Income ratio: " +
  the yearly repayment of 'the loan' / the yearly income of 'the borrower';
```

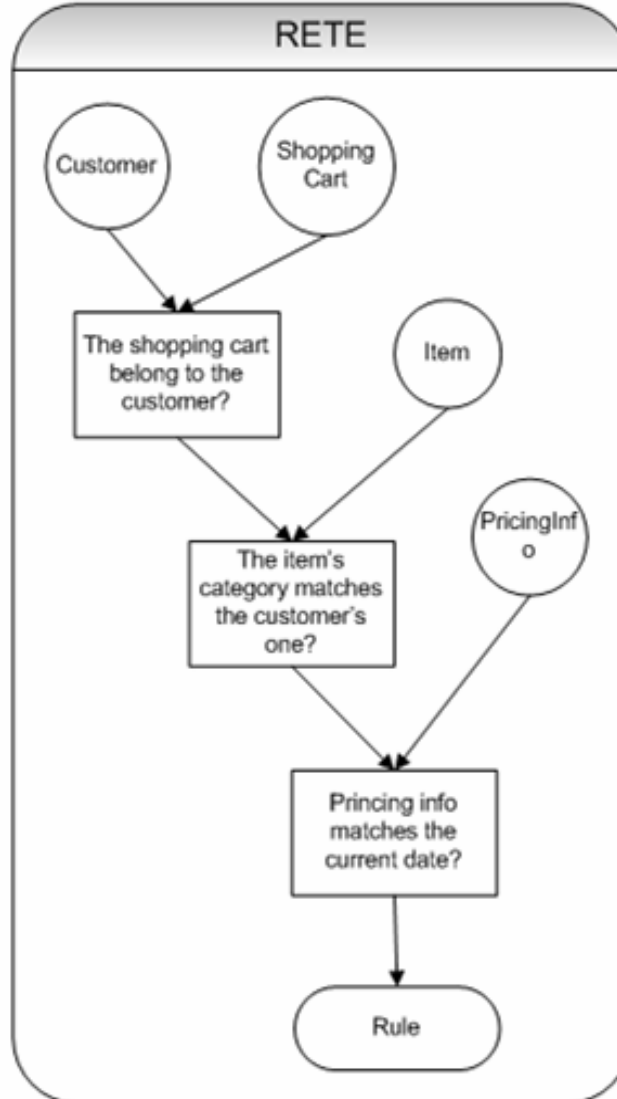
Rule Execution Algorithms



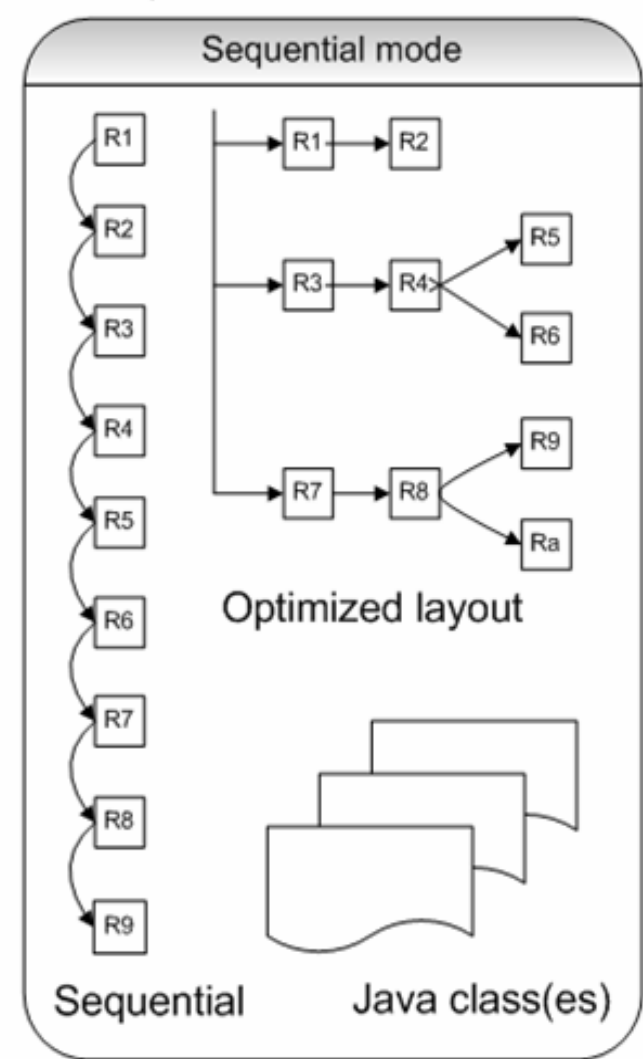
Task chaining



Computation, correlation rules



Compliance, validation rules

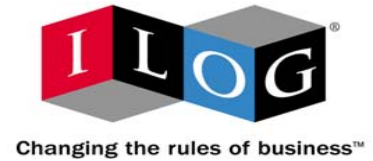


Challenges

- **Change frequency**
 - Business users write business rules
 - Frequently changed
 - Interpreters rather than compiled
- **Infer on user's objects**
 - Application area dependency
 - Classes are not known in advance
- **Standard platforms**
 - Written in Java
 - Integrated with J2EE
- **Scalability**
 - There can be numerous rules (hundreds of thousands)
 - There can be numerous objects to process

→ **The use of various optimization techniques**

Rule Execution



Typical activity

- **Evaluate the rules conditions**
 - Customer's category is gold
 - Shopping cart contains more than 3 CDs
 - There is at least an element in the collection such that ...
 - **Select the rules and execute the rules**
 - Set a 20% discount
 - Offer a free CD
- The interpreter infers on the customer's classes
- Read fields/invoke methods and compare with values
- Use Java introspection

Optimization 1

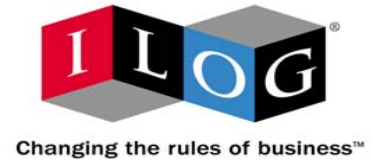


Changing the rules of business™

Challenges

- **Dynamic code generation at runtime**
 - The code that performs the tests are generated in bytecode
 - ➔ Adaptation to the user's classes
 - Special class loaders to load and execute the bytecode
 - Avoid using the Java introspection
 - Reduce the activity of the Java GC
- **Keep the dynamicity without paying for performance**
- **Compared to Java introspection: several times faster**

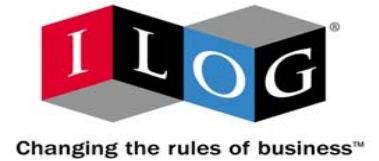
Optimization 2



Users know their classes and rules

- **Tell the rule engine that a field value is often used**
 - For example: customer's name
 - Rule engine will index on this field
 - → Build optimized structures
 - **Optimizing tests using specific data structures**
 - ==, >=, <= tests
 - Speed up the finding of objects some reference value
 - Binary trees, hashing algorithms
- **Options that allow customers to tell about their classes**

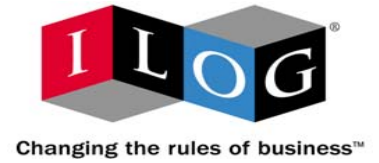
Optimization 3



Specific rule engines

- **Generating stand-alone programs**
 - **For “sequential rule processing” only**
 - **Specific application area**
 - **Compliance & validation rules**
- **The whole program is generated dynamically in Java bytecode**
- **Dynamically generated, loaded and executed**

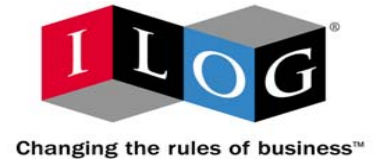
Other optimizations



Many others...

- Leave hooks
- Test sharing
- Caching
- ...

Conclusions



What has been adaptive?

- **Unknowns**
 - **User's classes**
 - **User's rules**
- **Requirements**
 - **Change frequency**
 - **Performance and scalability**
- **Adaptive behavior:**
 - **Optimize or generate code according to the classes and rules, at runtime**
 - **Keep the dynamicity and without losing performance**

Thank you!



- Any questions?