Integrating Business Rules and Constraint Programming Technologies for EDM

Dr. Jacob Feldman, OpenRules, Inc., Chief Technology Officer, jacobfeldman@openrules.com
j.feldman@4c.ucc.ie

Prof. Eugene Freuder, Cork Constraint Computation Centre, Director
e.freuder@4c.ucc.ie
Outline

- Business Optimization and Constraint Programming (CP)
- Real-world examples
- BR + CP integration scenarios
- How different CP solvers can be incorporated into different BR products: CP-Inside
- Online decision support with Business Rules, Constraints, and Machine Learning
Business Optimization

- Optimization usually refers to a mathematical technique used to calculate the best possible resource utilization to achieve a desired optimization objective such as:
  - minimizing expenses or travel time
  - maximizing ROI or service level, etc.

- Business Optimization helps business people to find optimal solutions among multiple alternatives subject to different business constraints

- Optimization Engine:
  - Determines how to most effectively allocate resources, automatically balancing trade-offs and business constraints
  - Eliminates the need to manually work out plans and schedules, so you can achieve maximum operational efficiency
Constraint Programming: a bridge between academy and business

- Constraint Programming (CP) is a proven optimization technology that is the most friendly to corporate developers.
- During the 90s CP successfully built a bridge between the academic and business worlds by providing an API for the mainstream programming languages.
- Constraints arise in design and configuration, planning and scheduling, diagnosis and testing, and in many other contexts.
- CP was successfully applied to solve real-world problems in:
  - telecommunications, internet commerce, electronics, bioinformatics, transportation, network management, supply chain management, finance, manufacturing, and many other fields.
Typical CP Applications

- Scheduling and Resource Allocation
- Complex Configuration Problems
- Supply Chain Management
- Staff Rostering
- Vehicle Routing
CP and Optimization are back

- In recent years CP was suppressed by BR
- Nowadays CP and Optimization technology in general is quickly becoming a “must-have” component of the EDM – Enterprise Decision Management
- Both BR leaders ILOG and Fair Isaac put Optimization among key components of their EDM vision:
  - ILOG has for a long time the best optimization tools including famous ILOG CPLEX and CP Optimizer
  - Fair Isaac recently acquired and incorporated Xpress-MP in their product offerings
- Many open source optimization products also achieved a competitive level and are ready for the prime time
- Incorporating CP into Business Rules Management systems empowers a BRMS with much more sophisticated decision-support capabilities
When CP does well

- **CP does well when BR stops short:**
  - Number of alternatives goes beyond thousands and millions
  - Close to optimal decisions are expected in real time
  - Compromise between time and quality is required

- **Powerful and affordable CP tools are available now:**
  - Open source CP Solvers
  - A choice of commercial CP Solvers
  - Friendly APIs for mainstream programming languages
Constraints and Rules are similar

Both rules and constraints represent conditions which restrict our freedom of decision:

- The meeting must start no later than 3:30PM
- Glass components cannot be placed in the same bin with copper components
- The job requires Joe or Jim but cannot use John
- Bob prefers not to work on weekends
- The portfolio cannot include more than 15% of technology stocks unless it includes at least 7% of utility stocks

Both rules and constraints support declarative programming

- Concentrate on WHAT instead of HOW
- The same basic idea:
  - a user states the Rules (or Constraints)
  - a general purpose Rule Engine (or Constraint Solver) solves them
Constraints and Rules are different

BR Advantage:

- Rules Repository is managed by business people while Constraints are usually under control of software developers

CP Advantage:

- Rules usually should consider all (!) possible combinations of the problem parameters
- Constraints do not have to cover all situations but rather define an optimization objective and allow a search algorithm to find the optimal solution

BR+CP provides the best of both worlds:

- BR defines an optimization problem, CP solves it
Financial Portfolio Management

Business rules such as “Technology Stocks should be within 15% and 20%” define the target portfolio

Rule Engine warns about possible rule violations during sell/buy

Business Objective:

- Keep all portfolios as close as possible to the target portfolio

Constraint Solver recommends the best combination of trade orders to minimize the total rules violation
Loan Origination

- A borrower provides a desired loan purpose, amount, term along with personal data
- Rule Engine executes complex eligibility rules to define all available loan products for this borrower

Business Objectives:
- avoid loan rejections or lengthy “what-iffing”
- allows “reasonable violations” of the requested loan parameters while recommending a loan with a minimal interest rate

Constraint Engine recommends the best combination of the loan amount, term, and product characteristics to select the most suitable loan product
Telecom Service Configuration

- Personalized configuration of available calling plans and other wireless, local, long distance, and Internet services
- Rule Engine determines cross/up selling opportunities and specifies different marketing campaigns
- Business Objective:
  - Customer retention
  - Using a customer’s preferences and an actual calling pattern recommend the best set of services while a CSR is on the phone with a complaining customer
- Rule and Constraint Engines work together with customer data to determine and deliver the best account management advice
Insurance Pricing Discount Calculation

- According to the specified business rules, the customer is eligible to \( N \) different discounts
- There is a rule/constraint that states that the total discount cannot be more than \( x\% \)
- Business Objective:
  - find a combination of the discounts that satisfies the “\( x\% \)” constraint while maximizing/minimizing the premium
- Rule engine figures out all eligible discounts
- Optimization engine finds the best alternative for customer and for the insurance company
Field Service Scheduling a public utility company

- Typical problem:
  - more than a million customers
  - several thousands employees with different skills and equipment
  - large service territory
  - hundreds or thousand small and large jobs per day
  - each job requires a mix of people skills, vehicles and equipment

- Conflicting Business Objectives:
  - Minimize travel time, level resource load, use the least costly skills/equipment, schedule jobs ASAP, honor employee preferences

- Rule Engine configures preferences and relative objectives for different constraint satisfaction problems

- Constraint Solver does on-demand scheduling and resource allocation
Business rules could be used to define and modify a business objects

Rule Engine can generate a related constraint satisfaction problem representing it in terms of constrained objects and constraints

CP Solver can solve the optimization problems and return the results to the Rules Engine for further analysis
BR + CP Integration Schema
Example: Sudoku with Rules & Constraints

Rules vs. Constraints:

- You may use rules to represent and solve the Sudoku problem. But you will end up with many very technical and hard to understand rules.

- OR you may incorporate the predefined constraint “AllDifferent” into a simple decision table (next slide)
Sudoku Constraints in Excel Rules Table

<table>
<thead>
<tr>
<th>Array Name</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>x00 - x08</td>
<td></td>
</tr>
<tr>
<td>x10 - x18</td>
<td></td>
</tr>
<tr>
<td>x20 - x28</td>
<td></td>
</tr>
<tr>
<td>x30 - x38</td>
<td></td>
</tr>
<tr>
<td>x40 - x48</td>
<td></td>
</tr>
<tr>
<td>x50 - x58</td>
<td></td>
</tr>
<tr>
<td>x60 - x68</td>
<td></td>
</tr>
<tr>
<td>x70 - x78</td>
<td></td>
</tr>
</tbody>
</table>

Rules void postSudokuConstraints(CpProblem p)

CpVariable[] array = p.addArray(name, vars);
p.allDiff(array).post();

<table>
<thead>
<tr>
<th>String name</th>
<th>String[] vars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array Name</td>
<td>Variables</td>
</tr>
<tr>
<td>block00</td>
<td>x00 x01 x02</td>
</tr>
<tr>
<td>block01</td>
<td>x03 x04 x05</td>
</tr>
<tr>
<td>block02</td>
<td>x06 x07 x08</td>
</tr>
<tr>
<td>block03</td>
<td>x10 x11 x12</td>
</tr>
<tr>
<td>block04</td>
<td>x13 x14 x15</td>
</tr>
<tr>
<td>block05</td>
<td>x16 x17 x18</td>
</tr>
<tr>
<td>block06</td>
<td>x19 x20 x21</td>
</tr>
<tr>
<td>block07</td>
<td>x22 x23 x24</td>
</tr>
<tr>
<td>block08</td>
<td>x25 x26 x27</td>
</tr>
<tr>
<td>block09</td>
<td>x28 x29 x30</td>
</tr>
<tr>
<td>block10</td>
<td>x31 x32 x33</td>
</tr>
<tr>
<td>block11</td>
<td>x34 x35 x36</td>
</tr>
<tr>
<td>block12</td>
<td>x37 x38 x39</td>
</tr>
<tr>
<td>block13</td>
<td>x40 x41 x42</td>
</tr>
<tr>
<td>block14</td>
<td>x43 x44 x45</td>
</tr>
<tr>
<td>block15</td>
<td>x46 x47 x48</td>
</tr>
<tr>
<td>block16</td>
<td>x49 x50 x51</td>
</tr>
<tr>
<td>block17</td>
<td>x52 x53 x54</td>
</tr>
<tr>
<td>block18</td>
<td>x55 x56 x57</td>
</tr>
<tr>
<td>block19</td>
<td>x58 x59 x60</td>
</tr>
<tr>
<td>block20</td>
<td>x61 x62 x63</td>
</tr>
<tr>
<td>block21</td>
<td>x64 x65 x66</td>
</tr>
<tr>
<td>block22</td>
<td>x67 x68 x69</td>
</tr>
<tr>
<td>block23</td>
<td>x70 x71 x72</td>
</tr>
<tr>
<td>block24</td>
<td>x73 x74 x75</td>
</tr>
<tr>
<td>block25</td>
<td>x76 x77 x78</td>
</tr>
<tr>
<td>block26</td>
<td>x79 x80 x81</td>
</tr>
<tr>
<td>block27</td>
<td>x82 x83 x84</td>
</tr>
<tr>
<td>block28</td>
<td>x85 x86 x87</td>
</tr>
<tr>
<td>block29</td>
<td>x88 x89 x90</td>
</tr>
</tbody>
</table>
Some Popular CP Tools

Java API
- Choco - Open Source (http://choco.sourceforge.net)
- ILOG JSolver – Commercial (www.ilog.com)

C++ API
- ILOG CP – Commercial (www.ilog.com)
- Gecode – Open Source (www.gecode.org)

CP environments with specialized modeling languages
- OPL from ILOG, France (www.ilog.com)
- MiniZinc from G12 group, Australia (http://www.g12.cs.mu.oz.au)
- Comet, Brown University (www.comet-online.org)
- Prolog-based tools (ECLiPSe, SICStus)

20+ other CP Solvers: http://slash.math.unipd.it/cp/

CP Solvers are usually well integrated with other optimization tools (LP, MIP)
Generic interface between different CP Solvers and Business Applications

Created by Cork Constraint Computation Centre (www.4C.ucc.ie) with support from Enterprise Ireland and Science Foundation Ireland

Provides a Vendor-Neutral CP API for Java

Adapters to popular open source and commercial CP solvers
Common library of constraints and goals
Standardization efforts (OMG)

Can incorporate CP-based engines in popular software tools:

MS Office (Excel), Rule Engines (OpenRules), Google Calendar and Facebook Events, MatLab, CEP tools, Lotus Notes, and others
CP-INSIDE Architecture (courtesy of www.4c.ucc.ie)

World of Business Applications

- Web Applications
- Interfaces to Popular Tools

Business Interfaces

- Constraint Solver Adapters

Common CP API

- Open Source Solvers
- Commercial Solvers

World of CP Professionals

- Constraint Library
- Search Goal Library

Different BRMS(s)

CP Add-Ons:
- Scheduler
- Configurator
- Recommender
Online Decision Support: modeling and solving constraint satisfaction problems

Typical Online Systems with CP-based Solvers:
- Online Reservation systems (hotels, tours, vacations, ..)
- Event Scheduling (both business and personal events in social networks)
- Field Service Scheduling, and more

Traditional Approach:
- “Fat” Problem Solver tuned for all possible problem states
- Complexity grows over time – hard to create and maintain
Online Decision Support: CP + BR
adding Rule Engine to find the “best” strategy

CSP (New State)

CP Solver
(the “best” strategy)

New Change Request

Predefined Models and Strategies

Rule Engine

Selected Model & Strategy

State Analysis and Strategy Selection Rules

IT Guru

Business Analysts
Online Decision Support: adding Rule Learner to find the “best” strategy

- **CSP (New State)**
- **CSP (Current State)**
- **CP Solver** (the “best” strategy)
- **New Change Request**
- **Selected Model & Strategy**
- **Predefined Models and Strategies**
- **Rule Engine**
- **Rule Learner**
- **State Analysis and Strategy Selection Rules**
- **Historical CSP States**
- **State Analyzer**
  - Positive & Negative Samples
Integration of BR and CP empowers a BRMS with much more sophisticated decision-support capabilities

BR+CP methodology and tools are available in a vendor-neutral way

Online decision support may be done with

- CP or BR only: Hard to create and maintain “fat” Solver controlled by IT
- CP + BR: Rule Engine recommends a CSP model and search strategy based on state analysis rules controlled by business analysts
- CP + BR + ML: Rule Learner discovers model/strategy selection rules based on historical Solver runs – “Ever-learning” decision support