OBJECT MANAGEMENT GROUP

A Unified Business Interface for Modeling and Solving Constraint Satisfaction Problems

Jacob Feldman, PhD
Cork Constraint Computation Centre, Sr. Researcher
www.4c.ucc.ie
OpenRules Inc., Chief Technology Officer
www.openrules.com

Cork Constraint Computation Centre

Outline

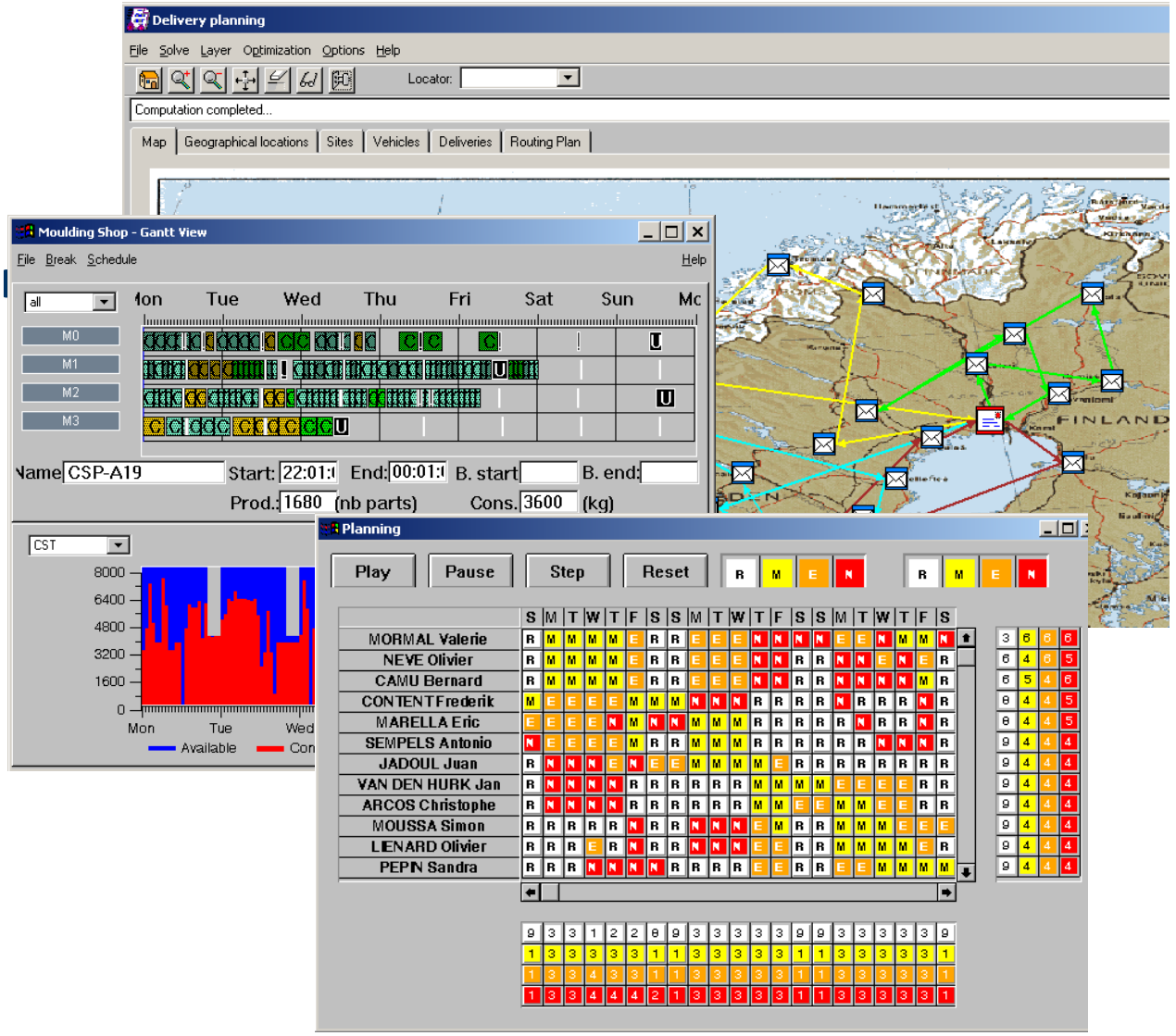
- **Optimization and Constraint Satisfaction Problems (CSP)**
  - Optimization problems and constraint-based decision services
  - Constraint Satisfaction Problems - simple memo
  - Current constraint modeling and programming languages
- **Developing a Unified Business Interface for CSPs**
  - Standardization Directions and Requirements
  - Problem Definition Concepts
  - Problem Resolution Concepts
- **Integration**
  - Integration with the existing CP Tools
  - Integration with OMG Standards

Optimization Problems

- **Optimization is at work everywhere: manufacturing, transportation, logistics, financial services, utilities, energy, telecommunications, government, defense, health care, retail, and social networks**
- **Optimization comes to play when:**
  - A business problem may have multiple solutions
  - We need to find one solution, all solutions , or an optimal solution that minimizes a certain business objective while satisfying different business constraints
- **Optimization problems can be extremely challenging computationally**
  - Cannot be solved in polynomial time
  - Modeling and solving optimization problems is an experimental endeavour: it is hard to predict if the model will work in practice
- **Over years many languages and tools have been designed to model and solve optimization problems**

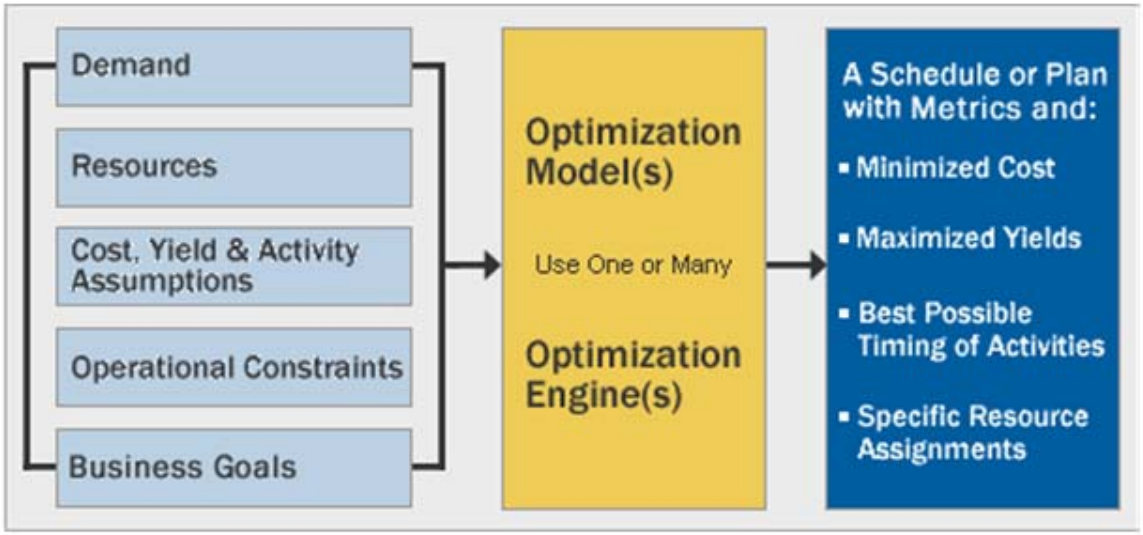
Typical Optimization Applications

- Scheduling and Resource Allocation
- Complex Configuration Proble
- Supply Chain Management
- Staff Rostering
- Vehicle Routing
- Enterprise Decision Management



Optimization Technology

(borrowed from www.ilog.com)



- Optimization technology helps organizations make better plans and schedules
- A model captures a complex planning or scheduling problem. Then a mathematical engine applies the model to a scenario and finds the best possible solution
- When optimization models are embedded in applications, planners and operations managers can perform what-if analysis, and compare different scenarios

Constraint Programming (CP)

- **Constraint Programming (CP) is a very powerful methodology for modeling and solving Optimization problems including non-linear ones**
- **The focus of CP is on reducing the search space by pruning values that cannot appear in any feasible or optimal solution. Constraints in CP are the main constructs that reduce the search space**
- **CP has deep roots in Operation Research and AI.** Useful links:
  - *Handbook of Constraint Programming* (Elsevier, 2006)
  - ACP - Association for CP - http://slash.math.unipd.it/acp/
- **During the 90s CP products such like ILOG Solver successfully built a bridge between the academic and business worlds. Today many CP Solvers empower regular business application developers with problem solving capabilities previously available only to AI gurus**
- **However, absence of standards still limits CP acceptance by business world**

CP Solvers are similar to Business Rules Engines

- **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 BR and CP 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

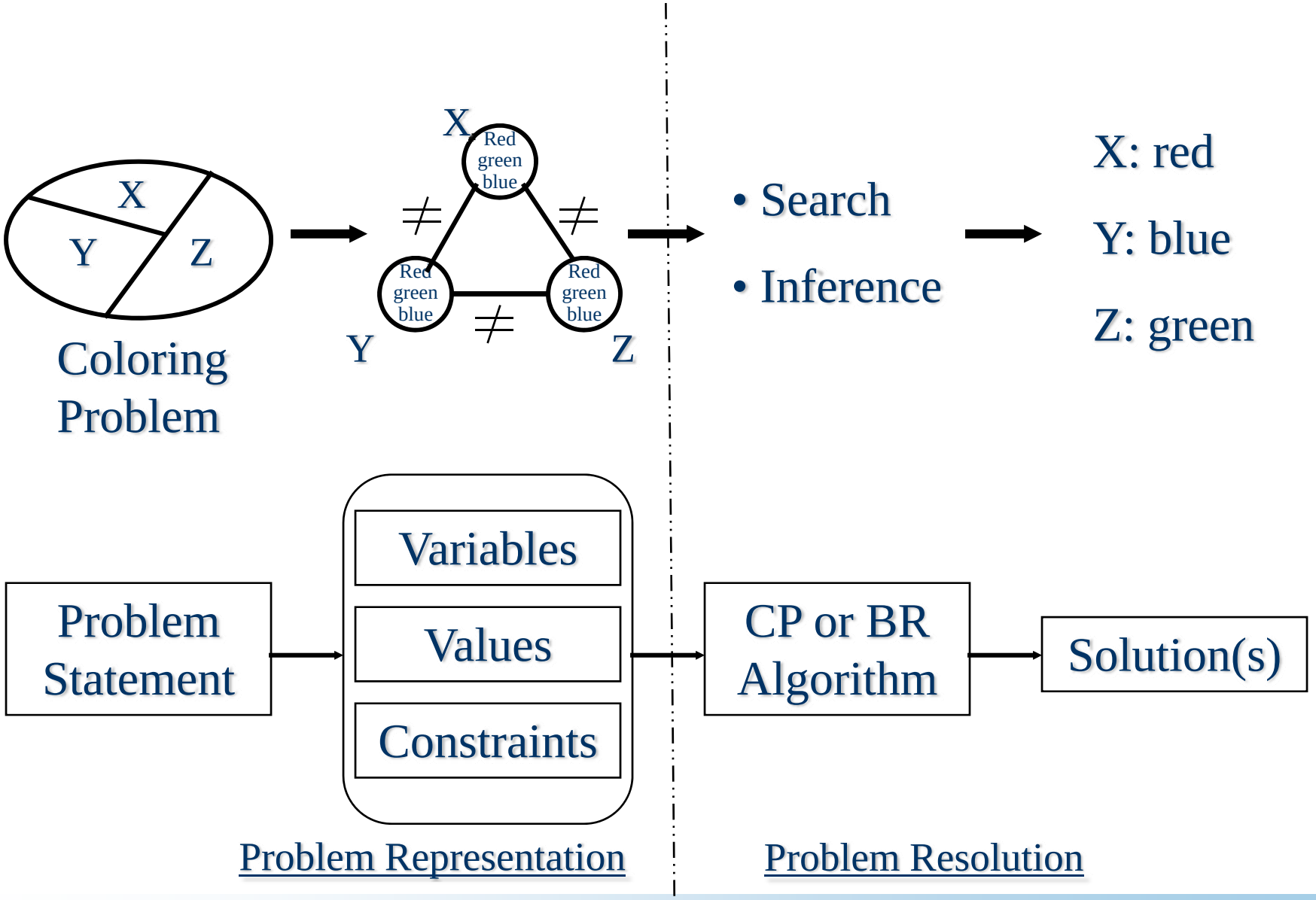
Constraints and Rules are different

- **Rules usually have to consider all (!) possible combinations of the problem parameters**
- **Constraints do not have to cover all situations but only key relationships between problem parameters. Defining an optimization objective, a user allows a Constraint Solver to find an optimal solution**
- **Instead of one major BR algorithm (Rete), CP can utilize many predefined search algorithms. CP also allows a user to specify problem-specific search heuristics**
- **BR+CP combination provides the best of both worlds:**
  - BR defines a business problem
  - CP solves a related Constraint Satisfaction Problem

Constraint Satisfaction Problems

- **CP models a business problem as a Constraint Satisfaction Problem (CSP)**
- **CSP Representation**:
  - Finite set of decision variables V_1, V_2, …,V_n (unknowns)
  - Each variable V_i has a non-empty domain D_i of possible values
  - Set of constraints restricting the values that the variables can take
- **CSP Resolution**:
  - Determining whether the CSP has a solution, that is a set of values for all variables that satisfy all constraints
  - Finding a solution
  - Finding all solutions
  - Finding an optimal solution that minimizes a objective variable
  - Finding all optimal solutions
  - Finding a “good” solution

Problem Representation and Resolution: trivial example



How the Constraint “X < Y” works

- **Let’s assume X and Y are defined on the domain [0,10]**
- **Initial constraint propagation after posting X<Y constraint:**

  X[0;9]

  Y[1;10]

- **Changes in X cause the changes in Y**

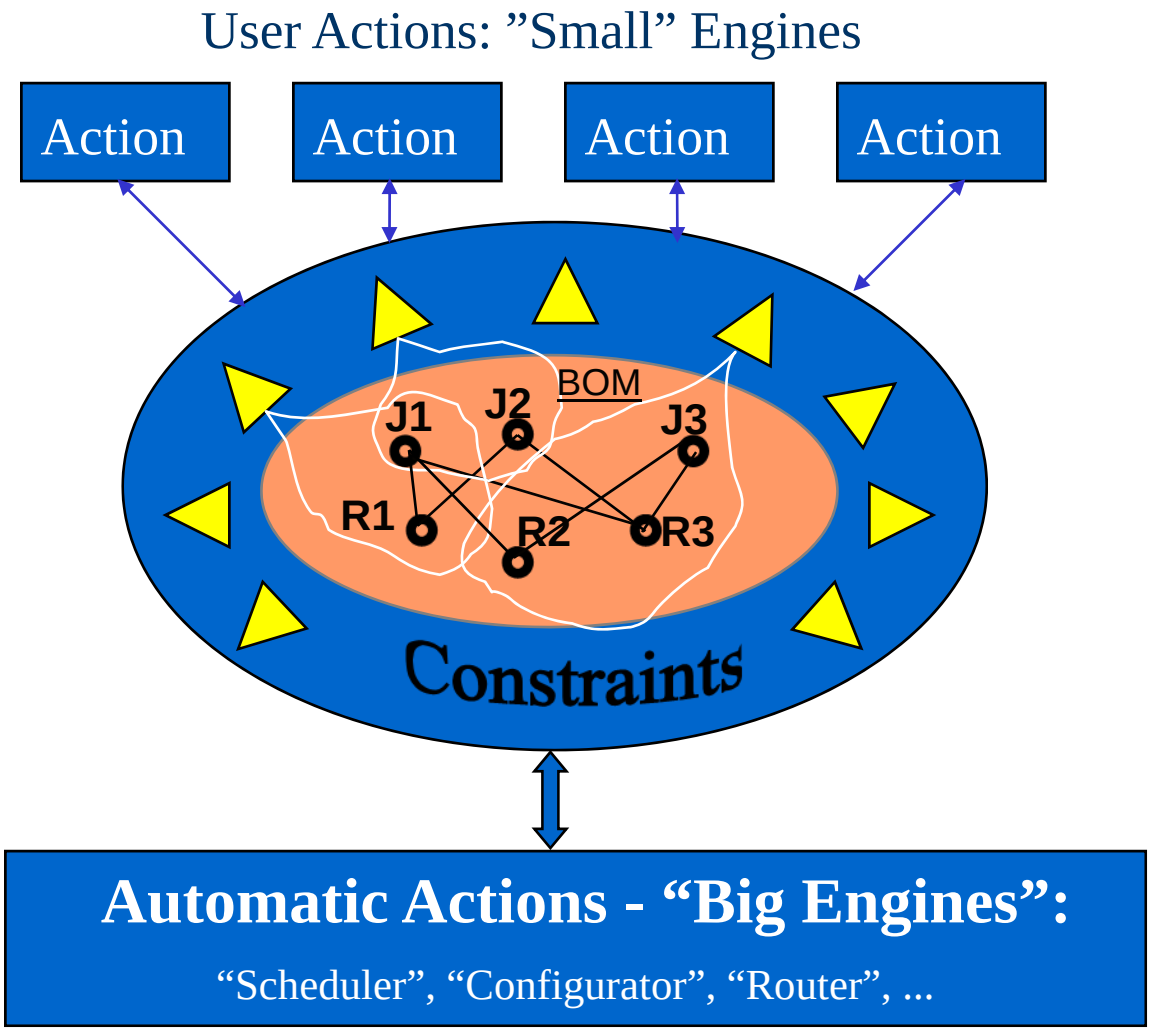
  X>3 => Y > 4

- **Changes in Y cause the changes in X**

  Y<=8 => X<=7

- **Bi-Directional constraint propagation**

Constraint Propagation (intuitive view)



Demo

Examples of CP frameworks

- **Specialized modeling and programming 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)
- **Main-stream programming languages with specialized CP APIs:**
  - **C++**

    ILOG CP – Commercial (www.ilog.com)

    Gecode – Open Source (www.gecode.org)
  - **Java**

    Choco - Open Source (http://choco.sourceforge.net)

    ILOG JSolver – Commercial (www.ilog.com)

    Koalog – Commercial (www.koalog.com)
- **Microsoft just introduced a MS Solver Foundation that includes a CP Solver**
  http://code.msdn.microsoft.com/solverfoundation
- **30+ other CP Solvers** (http://slash.math.unipd.it/cp)

Why CSP Modeling Needs to be Standardize

- **There are about 50 Constraint Solvers on the market today and all of them have their own unique CSP modeling facilities**
  - CSP modeling languages use varying levels of abstraction:
    some support global constraints, others support only a special input format and provide no help to a modeller
  - The CSP model often becomes dependent on the CP solver being used
  - Experimenting with different solvers requires learning each solver's modeling languages
  - Applying various solvers to the same model in reality means rewriting the model
  - Necessity to learn new proprietary languages often becomes a show-stopper for CP acceptance by business application developers
- **Many researchers world-wide use their own CP solvers**
  - Limits an ability to validate and compare valuable research results

CP is well-prepared for a unified Interface

- **Positive Unification Factors**
  - Clearly Defined Scope (common problem representation and problem resolution concepts)
  - Libraries of known CSPs
  - Recent unification initiatives
    - OPL
    - MiniZinc
    - CP-Inside
  - Support from several CP vendors and ACP
- **Problems to Overcome**
  - Vendor dependence
  - Absence of Common Vocabulary
  - Standardization should not limit vendor creativity
  - Simplification for business should not limit research
- **CP needs support from an established standardization body like OMG**

Standardization Directions

- **A unified business interface will support both processes:**
  - CSP modeling
  - CSP solving
- **Standardization will be based on a common vocabulary while supporting both implementation approaches:**
  - CP API for main-stream programming languages (Java , C#, C++)
  - Specialized CP modeling language (like OPL & MiniZinc)
- **To be considered:**
  - A standard XML schema for a CSP representation to serve as an interchange format

Standardization Requirements

- **Ability to use the same CSP model on different CP solvers**
- **Reference implementations for standard CSP interfaces:**
  - At least two reference implementations for existing CP solvers
  - Methodological recommendations of how CP solvers may implement the proposed standard interfaces
- **Standardized Libraries:**
  - Library Global Constraints
  - Library of popular CP problems with implementation models
- **Extensibility**
  - Ability to define custom constraints and search algorithms
  - Ability to extend the basic CSP model for different business verticals:
    - Scheduling and Resource Allocation Problems
    - Product Configuration Problems
    - Transportation Routing Problems and more
- **Simplicity and Expressiveness** (orientation for non-CP experts)

Major CSP Modeling Concepts

- **Problem Representation:**
  - **Constrained Variables** of different types:
    - Integer
    - Boolean
    - Real
    - Set
  - **Constraints**:
    - Basic arithmetic and logical constraints and expressions
    - Global constraints (defined on collections of variables)
    - Constraint Combinations
- **Problem Resolution:**
  - Search Goals (Algorithms)
  - Goal Combinations

Naming Convention

- **Names for all classes and methods are extremely important. A general principle: a user should be able to guess how to express constraints and search methods without looking into a reference manual**
- **The standard will define the default names for commonly used CP concepts**
  - For instance, "*Var*" may be used for the most popular type of constrained integer variables while other types could use additional qualifiers like "*VarReal*", "*VarSet*"
- **Allow synonyms wherever possible**
  - For instance, "*AllDiff*" may be used for the most popular global constraint but the name "*AllDifferent*" also should be allowed
  - Arithmetic operator "LessOrEqual" may be represented as "*x.lessOrEqual(y)*" but also as "*x.le(y)*", "*x.LE(y)*", or "x<=y", etc.
- **Naming conventions should be a subject for very serious discussions and a final decision should be made by a CSP Standardization Group with representatives from all major vendors**

Cooperation with CP Vendors

- **What not to standardize and leave to actual implementations**
  - Constraint propagation mechanisms
  - Domain implementation mechanisms for different domain types
  - Backtracking mechanism
  - Goal execution mechanism
  - Implementations of major binary and global constraints
- **Access to unique features from underlying CP Solvers**
  - The standardized model is supposed to be able to work with different underlying implementations
  - However, there should be loopholes that allow a user to violate this principle to take advantage of implementation specific features
- **OMG Certification**
  - Different implementations may be certified for compliance with the CSP modeling standard (even if they do not implement all features but provide the proper stubs)

Latest CSP Standardization Attempts - MiniZinc

- **MiniZinc is a CSP modeling language developed by a research group G12, Australia**
- **Reference:**
  - **Towards a standard CP modelling language**. N. Nethercote, P.J. Stuckey, R. Becket, S. Brand, G.J. Duck, G. Tack. MiniZinc: *Proceedings of CP-2007*, Providence, RI, 2007.
- **Interfaces to existing CP Solvers:**
  - ECLiPSe (Prolog)
  - GECODE (C++)

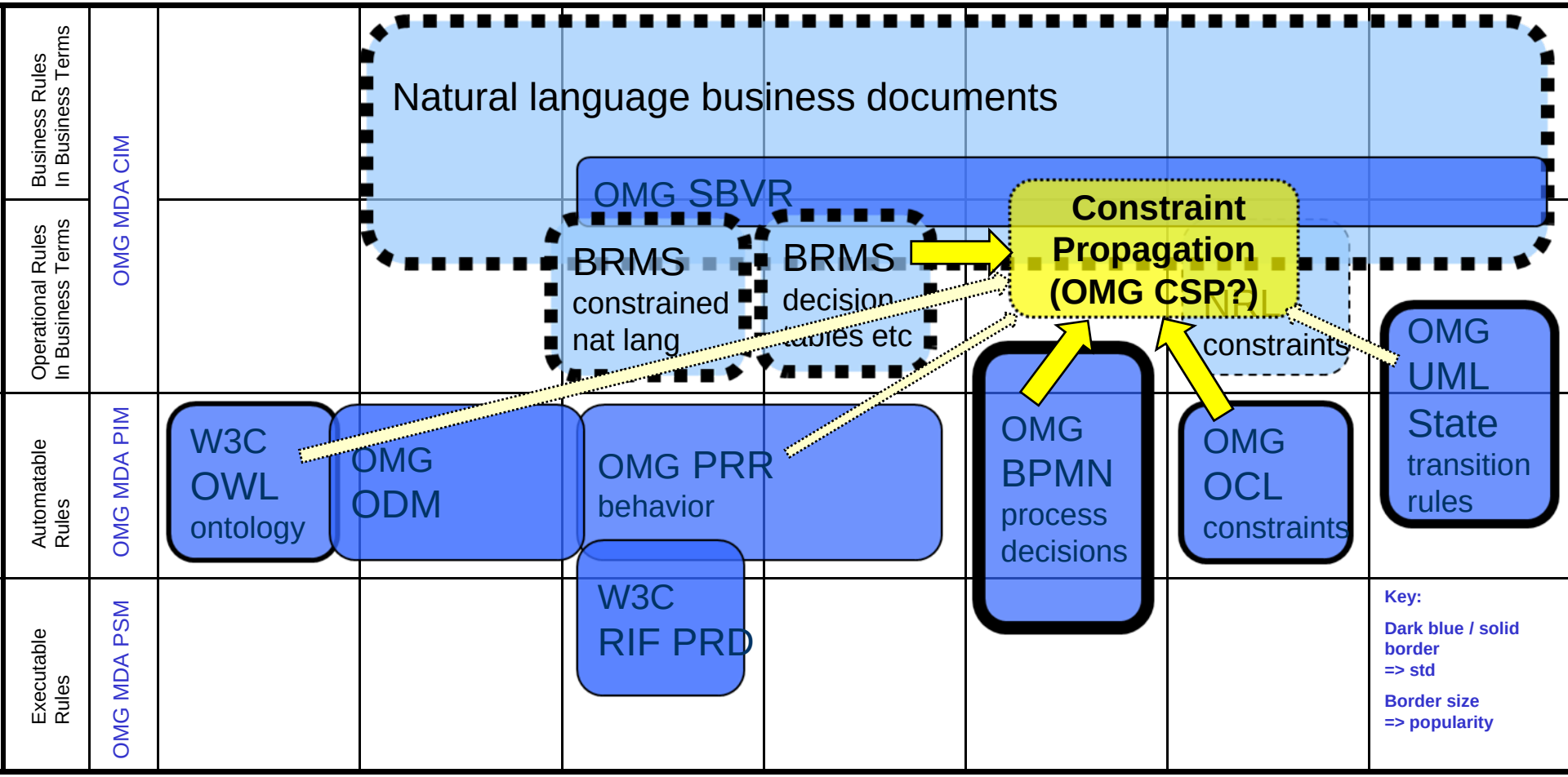
Latest CSP Standardization Attempts – CP-Inside

- **CP-Inside is a project developed by Cork Constraint Computation Centre, Ireland** (www.4C.ucc.ie)
- **Interfaces to existing CP Solvers:**
  - Choco (Java, open source)
  - ILOG JSolver (Java, commercial)
  - Constrainer (Java, open source)
- **Provides a Vendor-Neutral CP API for Java**
  - Adapters to popular open source and commercial CP solvers
  - Common library of constraints and goals
  - Scheduling Add-On
- **Provides interfaces to popular software tools:**
  - MS Office (Excel), Rule Engines (OpenRules), Google Calendar and Facebook Events, MatLab, and others

Where do Standards fit in Rules?

(courtesy of Paul Vincent, Tibco)

Where do Standards fit in Constraints?



OMG Standards and CSP

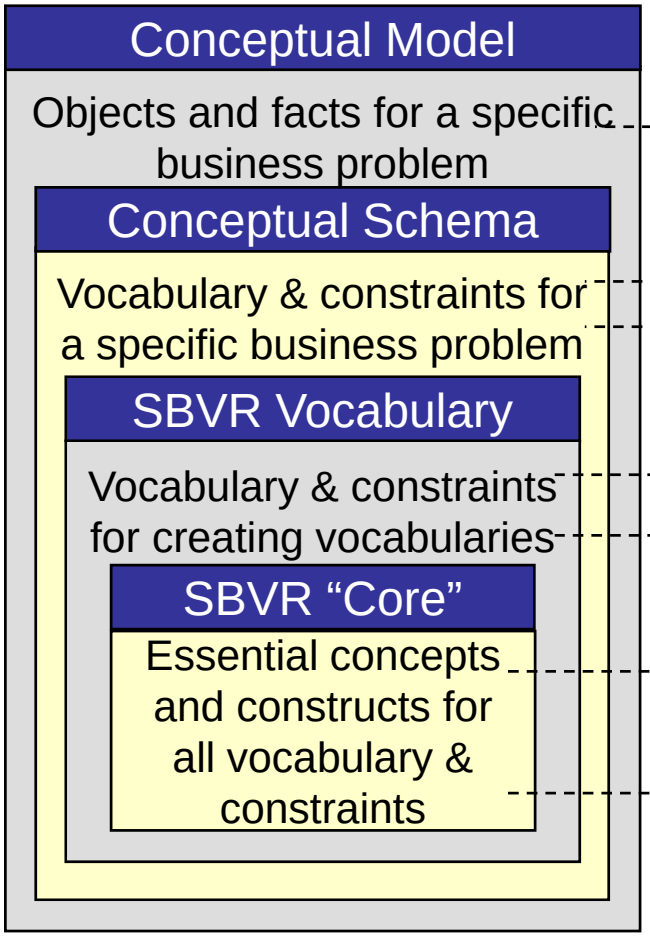
OMG Standard	CSP Connection
BPMN Business Process Modeling Notation	Process Decisions using Constraint-Based Decision Services
OCL Object Constraint Language	CSP is a specialize problem solving domain that may benefits from a more generic OCL specification
SBVR Semantics of Business Vocabulary and Business Rules	CSP Vocabulary can be designed as a specialized SBVR vocabulary oriented on modeling and solving optimization problems
PRR Production Rule Representation	CSP may help to automatically recommend optimal alternatives (behavior) when rules stop short
W3C OWL Semantic Ontology	CSP may be utilized to propagate new facts added to the existing ontologies

CSP Standard and OCL

- **OCL - Object Constraint Language**
  - A formal language used to describe expressions on UML models. These expressions typically specify invariant conditions that must hold for the system being modeled or queries over objects described in a model
  - OCL has a much broader scope than CSP and can be used
    - As a query language
    - To specify invariants on classes and types in the class model
    - To specify type invariant for Stereotypes
    - To describe pre- and post conditions on Operations and Methods
    - To describe Guards
    - To specify target (sets) for messages and actions
    - **To specify constraints on operations**
    - To specify derivation rules for attributes for any expression over a UML model.
- **CSP is a specialize domain that may benefits from a more generic OCL**
- **There were several attempts to generate CSP from OCL**
  - Example “UML to CSP”: http://gres.uoc.edu/UMLtoCSP/

Applying SBVR model to constraints

- Following the SBVR model described by John Hall at the OMG BMI meeting on Jan-2007:



Operative constraints:
"Carpentry takes 5 days and starts after masonry"
"Carpentry requires a carpenter"

"Activity requires resource"
"Activity starts after activity"
"Concept <activity> incorporates characteristics <start variable> and <duration>"

"Structural Constraints are based on fact types"

"Adopted definition is adopted from source vocabulary by community"

"Constrained variable incorporates a finite domain"

"There is a constraint "less than" between two constrained integer variables"

Summary “CSP Standardization”

- **Benefits:**
  - A unified vendor-independent CSP interface for business applications
  - Standardized library of constraints and business problems with predefined CSP models
  - Common Add-Ons for scheduling, configuration, routing, and other verticals
  - A unified way for scientists to present and validate their research results
- **Next Steps**
  - to be discussed