Constraint Solving with Advanced Variant Configuration

Andreas Koelbl, SAP
October, 2018
Agenda

Constraint Solving with Advanced Variant Configuration

• Introduction into AVC Constraint Solving
• Constraint Solving Example
• Implicit Constraints - Examples
• Demo: Solving Sudoku
• Summary
Introduction into AVC Constraint Solving
Configuration Engine

- Configurable Product
- Configuration Modeling
- Classification
- Sales Configuration
- Configuration Engine
- Configuration Simulation
- Bill of Material
- Routing
- Pricing Rules

Future Direction
Constraint Solving

Constraint
• Describes a condition that some variables must satisfy

Constraint Solving
• process of finding a solution to a set of constraints
• solution is the set of values for all variables that satisfies all constraints
Sequence of dependency processing

- Constraints
- Precondition on Value

- Interactive Assignments
- Static Defaults
- Procedures

- Unassign Defaults leading to inconsistency!

- Constraints
- Precondition on Value

- Precondition on Cstic
- Selection Condition on cstic
Constraint Solving Example
Example

Constraints

A = B
B < C
A ≠ D

Assignment

D = 5
C = 5
Event on Constraints

- Each constraint is triggered by an event
- Event depends on operator / syntax element

\[
\begin{align*}
A &= B & \rightarrow \text{Domain Changed} \\
B &< C & \rightarrow \text{Min / Max Value of Domain Changed} \\
A &\neq D & \rightarrow \text{Assignment Changed}
\end{align*}
\]
Complexity of Constraints

- Each constraints is assigned to a complexity cluster (from easy to crazy)
- Easy constraints are executed early, crazy constraints are executed latest

\[
\begin{align*}
A &= B & \rightarrow & \text{easy} \\
B &< C & \rightarrow & \text{easy} \\
A &\neq D & \rightarrow & \text{easy} \\
\end{align*}
\]

Read Table \rightarrow crazy
Restrict Solution - Example
Restrict Solution - Event Step 1.1

A
2 – 8

B
4 – 12

C
4 – 8

D
4 – 8

A = B
B < C
A ≠ D

Domain of A & B are different
Restrict Solution - Restrict Step 1.1

Create Intersection between A and B

A = B

A

2 – 8

B

4 – 12

4 - 8
Restrict Solution - Event Step 1.2

Event:
Max of B changed from 12 to 8
Restrict Solution - Restrict Step 1.2

B

\[ B = 4 \rightarrow 8 \]

C

\[ C = 4 \rightarrow 8 \]

Compare Min & Max Values

\[ B_{\text{max}} < C_{\text{max}} \]
\[ C_{\text{min}} > B_{\text{min}} \]
Restrict Solution - Event Step 1.3

Event:
Domain of B changed from 4 - 8 to 4 - 7
Restrict Solution - Restrict Step 1.3

Create Intersection between A and B
Restrict Solution - Result

A: 4 - 7
B: 4 - 7
C: 5 - 8
D: 4 - 8

A = B
B < C
A ≠ D
Post 1st Assignment & Restrict Solution
Restrict Solution – Assignment & Event Step 2.1

Event:
Assignment of D changed from unassigned to 5

Constraints

A = B
B < C
A \neq D

Assignment

D = 5
Exclude assigned value from unassigned domain
Restrict Solution – Event Step 2.2

**Constraints**

- $A = B$
- $B < C$
- $A \neq D$

**Assignment**

- $D = 5$

**Event:**

Domain of $A$ changed from $4 - 7$ to $4/6 - 7$
Restrict Solution - Result

Constraints
- A = B
- B < C
- A ≠ D

Assignment
- D = 5
Post 2\textsuperscript{nd} Assignment & Restrict Solution
Restrict Solution - Assignment

A: 4 / 6 -7
B: 4 / 6 -7
C: 5
D: 5

Constraints

A = B
B < C
A ≠ D

Event B < C:
Min/Max of C changed from 5 / 8 to 5
Result: B = 4

Event A = B:
Domain of B changed from 4 / 6 - 7 to 4
Result: A = 4

Assignment

D = 5
C = 5
Result

A  4
B  4
C  5
D  5

Constraints
A = B
B < C
A ≠ D

Assignment
D = 5
C = 5
Why you should use constraints...

Constraints

- restrict characteristic values to a feasible solution
- do not base on a general evaluation direction
- are not bound to a sequence
- are only executed if needed (principle of justification)
- discard inferences as soon as justification is not longer given

... and constraints are easy to maintain 😊
Implicit Constraints - Examples
Numeric characteristics w/o specific master data domain

VCH_DEMO_NO_DOMAIN < 100
VCH_DEMO_NO_DOMAIN ≥ 0

Demo No Master Data Domain (VCH_DEMO_NO_DOMAIN):
Value Range: 0 - 99
Condition in Constraints-Restriction

\[ X = 1 \quad \text{If} \quad Y = 2 \]

X is assigned to 2!

\[ 2 = 1 \quad \text{If} \quad Y = 2 \]

Valid solution enforces \( Y \neq 2 \)

Value 2 is excluded from domain of \( Y \)
Demo: Solving Sudoku
Summary
Summary

- Constraints are triggered by events
- Easy constraints are executed first, complex constraints are executed later
- AVC restricts the domain of the characteristics satisfying all constraints

Stay Consistent
Thank you.

Contact information:

Andreas Koelbl
Product Owner Advanced Variant Configuration Engine
andreas.koelbl@sap.com