Modelling and Solving Configuration Problems on Business Processes Using a Multi-Level Constraint Satisfaction Approach

Wolfgang Runte

University of Osnabrueck
Institute of Computer Science
Software Engineering Research Group
P.O. Box 4469, 49069 Osnabrueck, Germany

woru@informatik.uni-osnabrueck.de
http://www.inf.uos.de/se

Leipzig – SABRE 2009
Young Researchers Workshop Series on Modeling and Management of Business Processes (YRW-MBP'09)

March 25, 2009
1. Introduction

• Management of dependencies between business processes:
  − Problem: inconsistent process modellings.
  − Inconsistencies should be discovered in an early stage of modelling.
    • Reduce amount of time and cost.
  − Requirements of business processes depends on complex relations between the processes.
  − Usually the results of a foregoing process are needed by a subsequent one.

• Dependencies are relations between arbitrary attributes of business processes, examples are:
  − sequential dependencies
  − hierarchical dependencies
2. Consistent Configurations through Constraint Satisfaction

- Consistent configurations of business processes with methods out of the field of artificial intelligence (AI).
  - Knowledge-based configuration: using constraint satisfaction to model complex relations between (attributes of) components.

- Constraints as relations between attributes of processes:
  - algebraic constraints: intensional relations → equations/inequations
  - to reduce the possible assignments to variables (problem reduction)
  - for the (early) detection of inconsistencies
  - to generate solutions for a certain problem

- Constraint Satisfaction:
  - Characteristic: Propagation of changes throughout a “constraint net”.
  - Techniques for the handling of combinatorial and numerical problems.
Example: *sequential dependency*

- A constraint has to be satisfied in order that a process is allowed to be the successor of a foregoing process.

```
process 1  →  process 2

attribute: a  constraint pin: v₁ ← a  constraint relation(s): 0 < v₁ + v₂ < 10

constraint connector

attribute: b  constraint pin: v₂ ← b
```

process 1

```
3. Examples

Example: *hierarchical dependency*

- A constraint has to be defined to specify processes to be allowed to be nested sub-items of upper processes, in order to satisfy all requirements of super- and sub-processes.
4. Multi-Level Constraint Problem

• Goal: Handle different levels of nested business processes.

• Flexibility: Different layers of processes in hierarchies define different sub-problems.
  - the need to define different solutions strategies,
  - application of problem specific solving algorithms.

• For each sub-problem another solution strategy can be applied depending on:
  - the value domain of the involved variables,
  - the problem structure defined by the constraint net.

• Integration of local solutions of sub-processes has to be done on the higher-ordered level leading to global solutions and hence globally consistent configurations.
5. Summary

• Management of dependencies between business processes.

• Avoiding inconsistencies in business process modelling using constraint satisfaction.

• Constraints can be used to define arbitrary relations between attributes of business processes, e.g.
  - sequential and
  - hierarchical dependencies.

• Nested sub-problems on different abstraction levels:
  - can be seen as multi-level constraint problem,
  - results have to be integrated to upper levels for global solutions.
Thank you for your attention!
Outline of the talk:

1. Introduction
2. Consistent Configurations through Constraint Satisfaction
3. Examples
4. Multi-Level Constraint Problem
5. Summary
Sequential and Hierarchical Dependencies

... more precisely

- **sequential dependencies:**
  - Relations between processes in a sequential order.
  - Relations between the input/output values: the output of a foregoing process is needed as input of a subsequent process.

- **hierarchical dependencies:**
  - One or more processes can be sub-item(s) of a higher-ordered process.
  - Relations between lower and higher-ordered processes.
  - Relations between the input/output values of the first/last sub-process and the input/output of the higher-ordered process.
Constraints, Constraint Satisfaction Problem

- **Constraints** as relations between attributes of processes:
  - algebraic constraints: intensional relations → equations/inequations
  - to reduce the possible assignments to variables (problem reduction)
  - for the (early) detection of inconsistencies

- **Constraint Satisfaction Problem (CSP):**
  - Characteristic: Propagation of changes throughout a “constraint net”.
  - Techniques for the handling of combinatorial and numerical problems.
  - In the focus of intensive research and experiences for decades.
  - Efficient algorithms and heuristics:
    - reduction of the problem size/solution space
    - efficient generation of solutions
    - guarantee that specific relations hold
A Constraint Satisfaction Problem (CSP) is a triple $CSP(V,D,C)$:

$V = \{v_1, \ldots, v_n\}$ a finite set of variables

$D = \{D_1, \ldots, D_n\}$ associated value domains \{v_1 : D_1, \ldots, v_n : D_n\}

$C$ a finite set of constraints $c_i(V_i)$, $i \in \{1, \ldots, m\}$,

$c_i(V_i)$ to set the subset $V_i = \{v_{i_1}, \ldots, v_{i_k}\} \subseteq V$ in relation,

solution space for $c_i(V_i)$: $\{D_{i_1} \times \ldots \times D_{i_k}\}$

Example:

- Variables: $a$ and $b$ each with the value domain \{0,1,2,3,4,5,6,7,8,9\}
- Constraints: $a + b = 10$ and $a - b = 2$
- Solution: $a = 6$ and $b = 4$
- Note: Besides arithmetic domains also symbolic domains are feasible.
Example of a constraint graph: *map coloring problem*

- **Nodes** → Constraint variables
- **Edges** → Constraints

A possible solution for this CSP:

- $X = \{\text{red, green, blue}\}$
- $Y = \{\text{red, green, blue}\}$
- $Z = \{\text{red, green, blue}\}$
Modelling and Solving Configuration Problems on Business Processes Using a Multi-Level Constraint Satisfaction Approach

Static and Dynamic Usage of Constraints

- Usage of constraint relations for business processes:
  - static use → during modelling
  - dynamic use → during execution

  - Static use during modelling:
    - constraints connect input/output variables or attributes of processes
    - test for solutions and/or inconsistencies of the static model
      - Example: $a > b; a = [0..4], b = [5..9] \rightarrow$ inconsistent model

  - Dynamic use during execution:
    - test for solutions and/or inconsistencies during the execution of the business processes
    - user input or calculation results lead to reduced solution space
      - Example: $a \geq b; a = [0..9], b = [0..9] \rightarrow$ user input: $b = 5 \rightarrow a = [5..9], b = 5$