Formalizing Dependencies in Business Process Models Using Constraint Satisfaction

Wolfgang Runte

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

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

Leipzig – INFORMATIK 2010
Workshop on Protocol based modelling
of business interactions (PBMBI-GI'10)

September 30, 2010

Outline of the talk:

1. Introduction
2. Consistent Business Processes with Constraint Satisfaction
3. Static and Dynamic use of Constraints
4. Examples
5. Summary & Outlook

Outline of the talk:

1. **Introduction**
2. Consistent Business Processes with Constraint Satisfaction
3. Static and Dynamic Use of Constraints
4. Examples
5. Summary & Outlook

1. Introduction

- Management of dependencies in business process models
  - Modelling process models is a task of permanent change.
  - Like all modeling tasks errors may occur by human failure.
  - Problems:
    - Process models are incomplete.
    - Different modelers acting on the same process model.
  - Result: Process models may be inconsistent.
  - Inconsistencies should be discovered in an early stage of modelling.
    - Reduction in time and cost for process maintenance.

UNIVERSITÄT OSNABRÜCK

1. Introduction

- Quality assurance (QA) for business process models:
  - simulation of process models
  - defining input/output data for running tests
  - detecting inconsistencies during simulation should be efficient ...

- Inconsistencies during live operations should result in:
  - notification and/or
  - pre-defined regular operations to deal with the inconsistency

1. Introduction

- Idea to handle inconsistencies:
  - Modeling known dependencies into the process model.
  - A solver component will efficiently check this dependencies.
- During modeling, simulation and in live operations the solver component may alert in case of inconsistencies.
- Idea:
  - Formalizing the dependencies in business processes explicit as constraints.
  - Solver component will use *Constraint Satisfaction* to check for inconsistencies.

Outline of the talk:

1. Introduction
2. **Consistent Business Processes with Constraint Satisfaction**
3. Static and Dynamic Use of Constraints
4. Examples
5. Summary & Outlook

2. Consistent Business Processes with Constraint Satisfaction

- Check for consistencies in 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:
  - for the (early) detection of inconsistencies
  - algebraic constraints: intensional relations → equations/inequations
  - temporal operators to define temporal relations between processes
  - any other operator who seems to be useful ...

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

2. Consistent Business Processes with Constraint Satisfaction

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*** $c_i(V_i)$, $i \in \{1, \ldots, m\}$, *with*

$c_i(V_i)$ *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.

UNIVERSITÄT OSNABRÜCK

Outline of the talk:

1. Introduction
2. Consistent Business Processes with Constraint Satisfaction
3. **Static and Dynamic Use of Constraints**
4. Examples
5. Summary & Outlook

3. Static and Dynamic Use of Constraints

- Usage of constraint relations for business processes:
  - static use of constraints → at modeling
  - constraint solver will check for a consistent process model
  - dynamic use of constraints → at runtime (simulation or live)
  - constraint solver will check for consistent states of process instances

UNIVERSITÄT OSNABRÜCK

3. Static and Dynamic Use of Constraints

- Static use of constraints at modeling:
  - The modeler defines known static dependencies in the process model as constraints.
  - A constraint solver component checks efficiently for inconsistencies.
  - If a constraints does not hold the (current) modeler will get a notification.
- Dynamic use of constraints at runtime (simulation or live):
  - Known dependencies have to be defined in the process model.
  - Constraint solver checks for inconsistencies during the execution of the business processes.
  - Inconsistencies will result in pre-defined regular operations.

Outline of the talk:

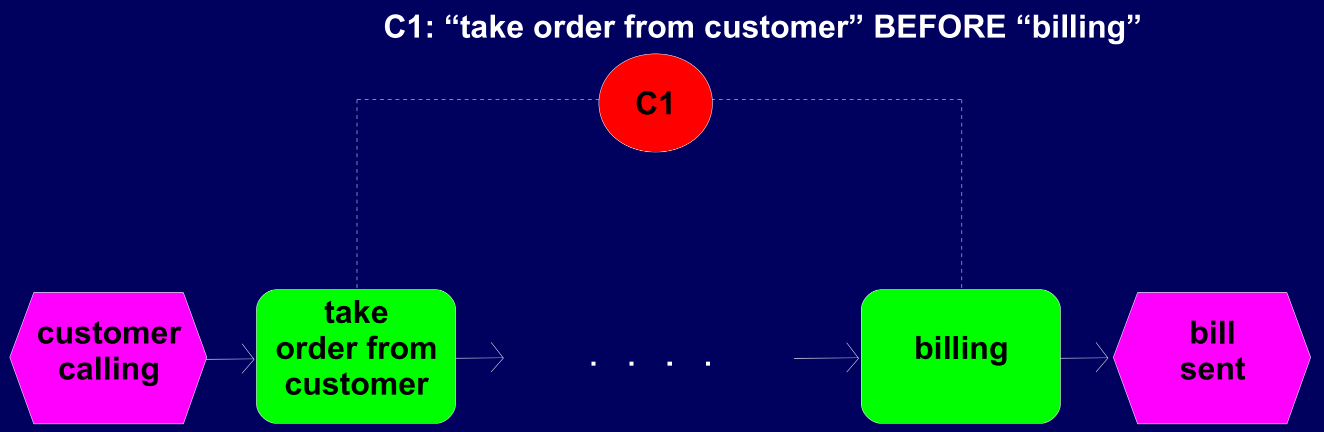
1. Introduction
2. Consistent Business Processes with Constraint Satisfaction
3. Static and Dynamic Use of Constraints
4. **Examples**
5. Summary & Outlook

4. Examples: static dependencies

- ***static structure constraints***:
  - These constraints are related to the static structure of a process model.
  - Examples:
    - A particular region in the process model does not allow cycles.
    - A particular ordering has to be kept for specific processes (e.g. process “A” has to be done before process “B” is able to start).
    - Particular processes have to get the same number of input values (e.g. they are related to the same items).
    - Check the value domains of input/output data (e.g. “A + B = C” may be result in values for variable “C” that are not in the value domain of “C”).

4. Examples: static dependencies

- Graphical example for ***static structure constraint***: (particular ordering)



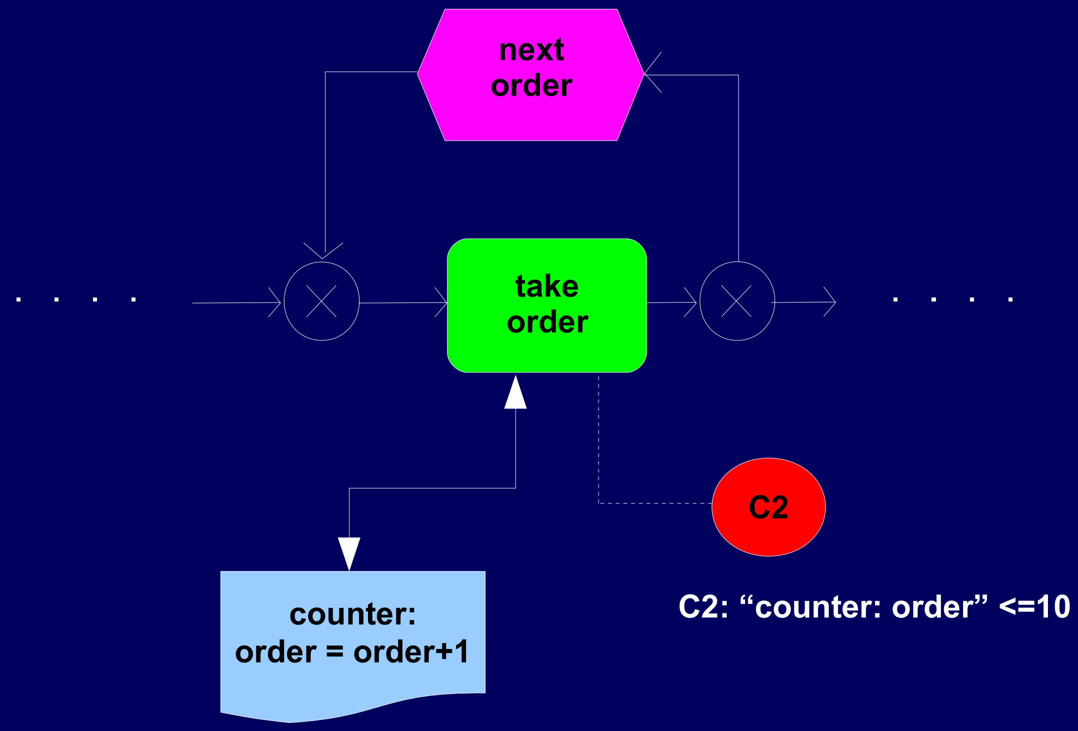
  - The constraint C1 only holds if process “take order from customer” is modelled *before* the process “billing” in the static process model.

4. Examples: dynamic dependencies

- ***dynamic structure constraints***:
  - Also related to the structure of the process model.
  - Because these constraints are checked during runtime they are more flexible than static structure constraints.
  - Examples:
    - Cycles can be restricted by counters, e.g. passing a process instance during runtime will increase a counter variable which is part of a constraint. By setting a maximum value for the counter it is possible to regulate these passing to a maximum number.
    - Particular processes have to get dynamically the same number of input values which are statically not known (e.g. they are related to the same bill of cost during billing).

4. Examples: dynamic dependencies

- Graphical example for ***dynamic structure constraint***: (restricted cycles)



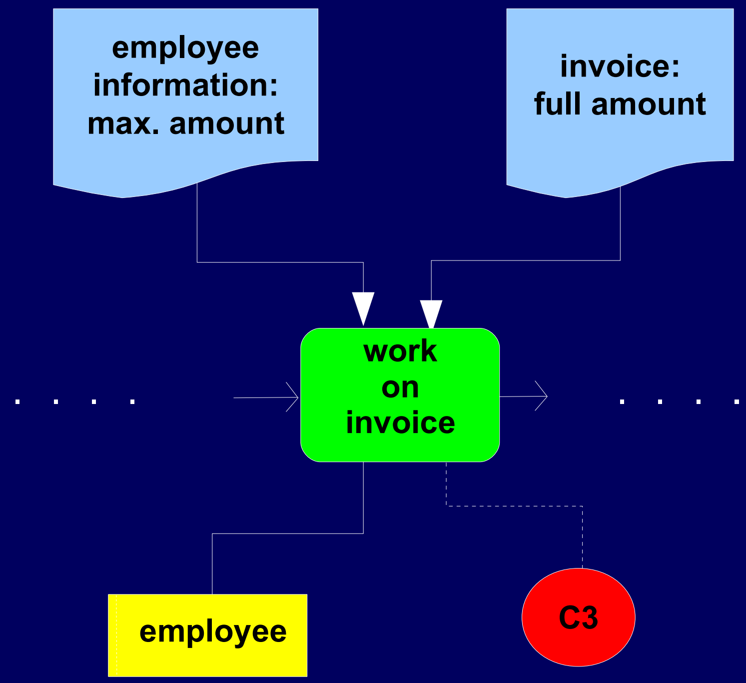
  - Constraint C2 defines that a maximum number of 10 orders is allowed.

4. Examples: dynamic dependencies

- ***input/output constraints***:
  - These constraints are related to the input and output values of a process instance.
  - Examples:
    - Quality assurance: Only items with the correct invoice number are included.
    - The full amount of the invoice has to be below a specific value (to be executed by a specific employee).

4. Examples: dynamic dependencies

- Graphical example for ***input/output constraint***:
  (value check)



C3: “invoice: full amount” <= “employee information: max amount”

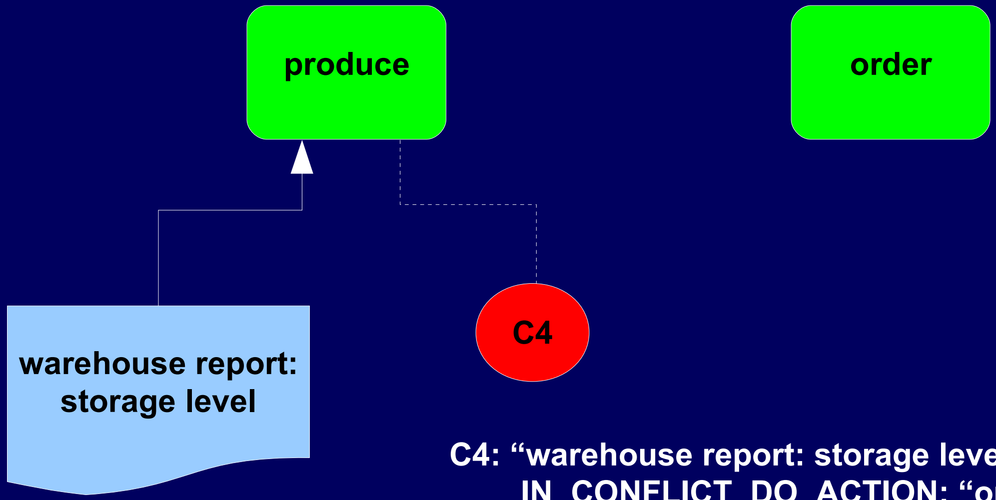
  – Constraint C3 holds if the employee is allowed to work on the invoice.

4. Examples: dynamic dependencies

- ***resource constraints***:
  - These constraints are dedicated for *continuous processes* respectively *continuous production processes*.
  - During continuous processes a permanent throughput is reached with specific resources.
  - Continuous processes are usually not embedded in a specific control flow, instead they are controlled indirectly.
  - If a monitored resource decreases below a defined value or raises over a defined value a specific action is taken.
  - Example:
    - If the storage level of a warehouse decreases below a defined value, an ordering is triggered to get new stuff.

4. Examples: dynamic dependencies

- Graphical example for ***resource constraint***:
  (value check)



C4: “warehouse report: storage level” >= 10%
IN_CONFLICT_DO_ACTION: “order”

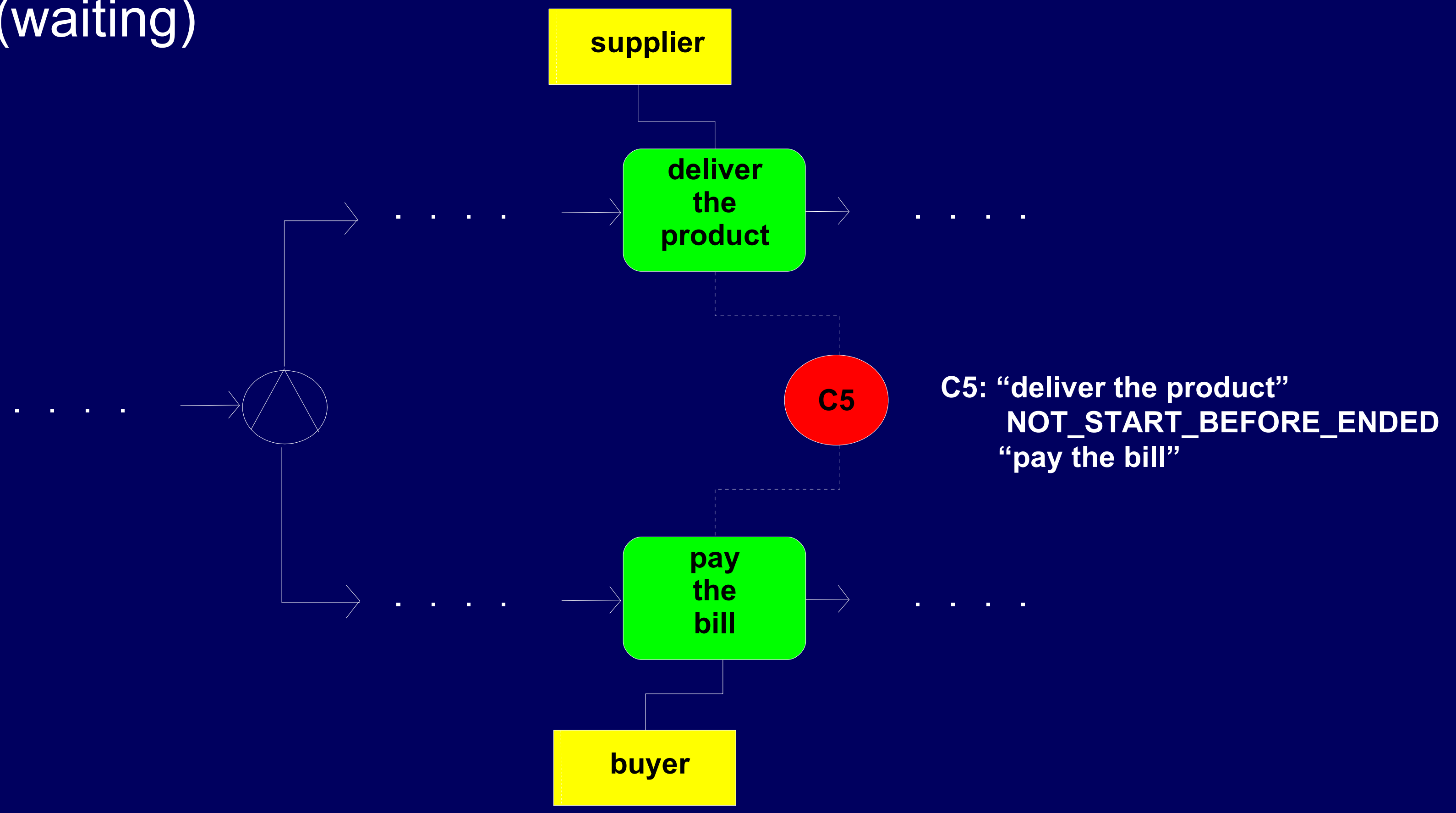
4. Examples: dynamic dependencies

- ***temporal constraints***:
  - Like rules in planning to plan actions temporal constraints focus on the time flow and can be used for scheduling.
  - *Temporal Constraint Satisfaction Problem* (TCSP)
  - Uses its own temporal logic representing temporal relations.
  - Here: TCSP is used for flow control instead of scheduling.
  - Example:
    - A process has to be waiting until another (parallel running) process in another branch of the process model has been ended.

Formalizing Dependencies in Business Process Models Using Constraint Satisfaction

4. Examples: dynamic dependencies

- Graphical example for ***temporal constraint***:
  (waiting)



  – Constraint C5 prevents process “deliver the product” to start before the process “pay the bill” has ended.

Outline of the talk:

1. Introduction
2. Consistent Business Processes with Constraint Satisfaction
3. Static and Dynamic use of Constraints
4. Examples
5. **Summary & Outlook**

5. Summary

- Management of dependencies in business process models.
- Avoiding inconsistencies in business process modeling using constraint satisfaction (static use of constraints).
- Avoiding inconsistencies during runtime (simulation or live operation) using constraint satisfaction (dynamic use of constraints).
- Categories of constraints identified representing different dependencies:
  - static:
    - structure constraints
  - dynamic:
    - structure constraints
    - input/output constraints
    - resource constraints
    - temporal constraints

5. Outlook

- Case study to identify some more categories and examples of dependencies.
- Study the adequacy of different CSP techniques for the different categories of business process dependencies.
- Evaluate business process modeling tools for the integration of consistency checking constraints solvers.
- Implementation of a prototype.

Thank you for your attention!

Questions, Ideas, Suggestions?

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

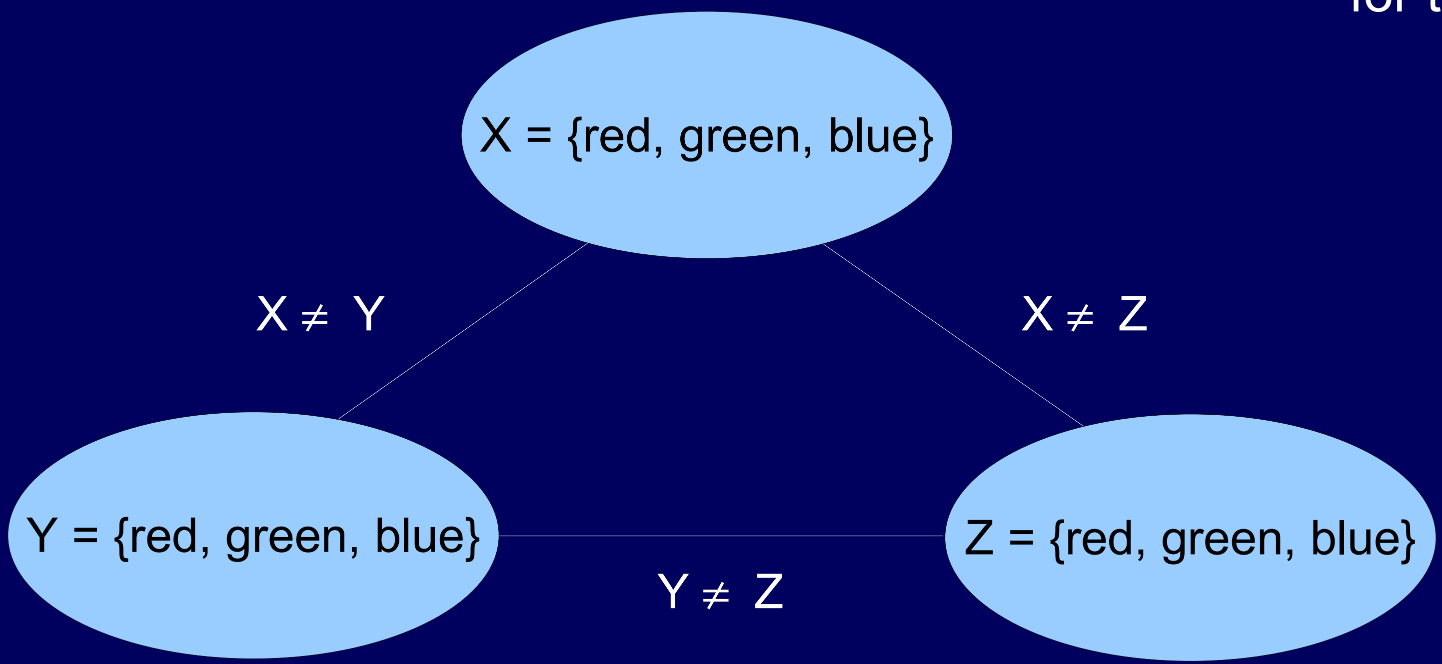
- *Constraint Satisfaction Problem* (CSP):
  - Characteristic: Propagation of changes throughout a “constraint net”.
  - Techniques for the handling of combinatorial and numerical problems.
  - In the focus of intensive research and experiences for decades.
  - Efficient algorithms and heuristics:
 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

UNIVERSITÄT OSNABRÜCK

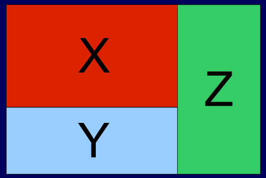
Constraints, Constraint Satisfaction Problem

Example of a constraint graph: *map colouring problem*

nodes → constraint variables
edges → constraints



A possible solution for this CSP:



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.