Constraint Checking for Business Process Management

Wolfgang Runte[1♣], Marwane El Kharbili[2♣]

[1] Institute of Computer Science, University of Osnabrueck
Albrechtstr. 28, 49076 Osnabrueck, Germany
wolfgang.runte@informatik.uni-osnabrueck.de
[2] ARIS Research, IDS Scheer AG
Altenkesselerstr. 17, 66115 Saarbruecken, Germany
marwane.elkharbili@ids-scheer.com

Abstract: Compliance management tackles issues related to both modelling and enforcement of business constraints in enterprises. In the context of business process management, we propose and describe the use of constraint satisfaction problems as a formal mean for representing these business constraints. We propose a multi-level constraint satisfaction approach to handle different levels of abstraction in business process modelling. We elaborate on the modelling of compliance requirements using constraint satisfaction and discuss implications of this technique such as problem solving strategies. This work shows that a certain class of compliance problems which can be grounded to constraint satisfaction problems can be solved efficiently using the proposed approach.

1 Introduction

Business Process Management (BPM) is the discipline of capturing, modelling, implementing, and controlling all activities taking place in an environment defining the enterprise, and this, in an integrated manner [Sch00, AHW03]. Several languages, frameworks, and tools that support one or many of the listed aspects are available. Organizations do not only own business processes, they are also subject to regulations. Companies are forced to implement measures ensuring they are compliant with these regulations, since non-compliance can cause judiciary pursuits or loss of market confidence and thus diminution in share value [KS+08]. Doing this, companies face two main problems: (i) finding suitable *modelling formalisms and languages for expressing* compliance constraints, while dealing with the *complexity* introduced by the various kinds of *interdependencies between business processes*. And (ii), proposing *efficient means of compliance verification and enforcement* that can both provide exact answers about the compliance status of business processes and this in a reasonable amount of time.

♣ Both authors are first authors and contributed equally to this work.

Among the various techniques already proposed for dealing with business constraints, modelling as *constraint satisfaction problems* (CSPs) is an attractive approach [Run09]. This position paper presents our ideas dealing with compliance in the context of *business process management* (BPM).

In the following, we elaborate on compliance in BPM and focus on the use of CSP for this purpose in Section 2. Section 3 contains an explanation of the approach proposed in this work, while Section 4 provides an example of constraint checking as we propose it for BPM. An overview of related work is given in Section 5. Finally, Section 6 contains concluding remarks together with planned future work.

2 Problem

2.1 On Compliance as Constraint Enforcement

Compliance means ensuring that structural and behavioural properties of business processes (BPs) (resp. business process architectures) are defined according to certain requirements. In the case of regulatory compliance, the latter requirements are provided by regulatory documents. Usually compliance requirements regard temporal properties of BPs, task selection criteria or resource allocation criteria as in [LS+06]. However, additionally to these requirements and to the dependencies between tasks in business processes, another aspect adds to the complexity of compliance management when observing business process in the context of BP architectures: *dependencies between processes*. In dealing with business process structures and compositions (which define architectures), *inconsistencies* may arise with regard to initial requirements on BPs as well as to compliance requirements. The problem we need to deal with is to (i) *find out about these inconsistencies* and to (ii) *be able to propose configurations* of BPs and BP architectures where these inconsistencies do not appear.

CSPs take a set of variables and constraints on these variables and seek a solution to the problem of finding a consistent assignment to the values of these variables [Kum92]. If it is possible to represent compliance constraints as CSP problems by finding out which BPs and BP architecture variables and constraints relate on them (i. e. relations describing the dependencies between BPs), we may transform the problem of compliance to a CSP problem. This work examines cases where this reduction makes sense and where CSP can be a satisfying tool for answering the question of compliance. When related back to the problem definition above, the problem we face is to be able to find configurations in terms of variable domains that ensure that no inconsistencies in BPs and architectures can occur. One example of dependencies between business processes are sequential dependencies (i. e. ordering), which can be represented as constraint sets on values of input/output variables of processes that are executed in a sequence. There may be other kinds of dependencies as well, such as parallel execution dependencies, synchronisation dependencies or vertical dependencies (i. e. compositions of business processes where a BP task is in fact implemented by a BP).

2.2 Constraint Enforcement for Business Processes

Constraint satisfaction problems have been in the focus of intensive research and experiences for decades. There exist efficient algorithms and heuristics for the reduction of problem size and for efficient generation of solutions. Overall, the techniques can be used to guarantee that specific relations hold [Kum92, Dec03]. A CSP is a triple (V, D, C) where V denotes a finite set of variables, D denotes a set of associated domains with possible values for each variable, and C denotes a set of constraints:

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

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

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

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

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

Each constraint defines a relation between a subset of variables and constrains the possible values for the involved variables. A constraint concerning only one variable is a unary constraint, constraints concerning two variables are binary constraints, constraints with three variables are ternary constraints, etc.

A short example would be the following simple constraint problem: Let there be two variables a and b each with the assigned value domain {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, and the binary constraints c_1: a + b = 10 and c_2: a - b = 2. The solution for this simple constraint problem would be a = 6 and b = 4. Note that besides arithmetic domains also symbolic domains are feasible and that the principles are not restricted to discrete domains.

Another widely used example is the map colouring problem: In figure 1 we see a small map consisting out of three parts x, y, and z each with the associated value domain {red, green, blue}. To generate solutions for the problem each part of the map has to be coloured in a way that direct neighbours have different colours.

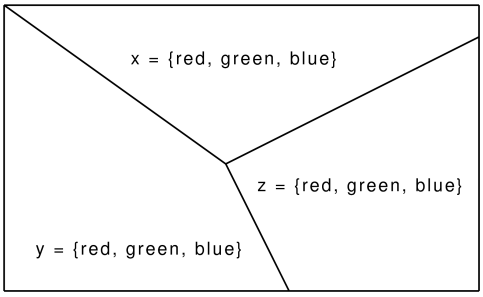

Figure 1: Graph colouring problem

The resulting constraint problem is represented by the constraint graph in figure 2: The constraint variables are represented as nodes, the edges of the graph are constraints between these variables. Possible solutions for this CSP would be: (1) x = red, y = green, z = blue, (2) x = green, y = blue, z = red, (3) x = blue, y = red, z = green, ... etc.

If for any reason the domain of a variable is reduced to a smaller number of values, this domain modification can be propagated through the constraint net using the available constraints to determine smaller value domains for the rest of the involved variables.

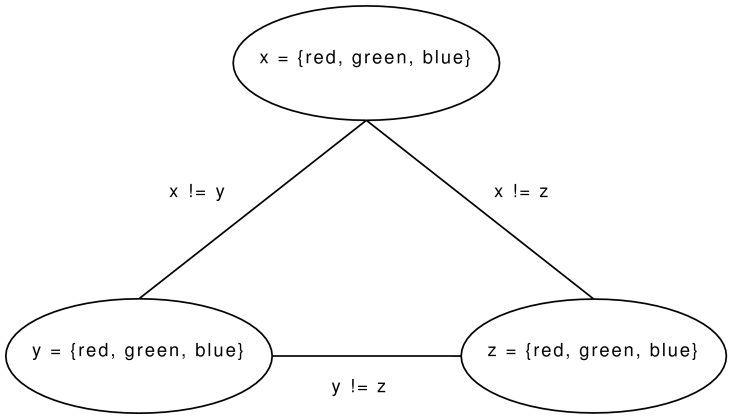

Figure 2: Constraint graph

For Example: If the domain of x in figure 2 is restricted to a single value, this value has to be removed from the domains of y and z because of the constraints between the involved variables. If further the domain of y or z is reduced by another value the respective domain is single-valued, too. In another propagation step this domain modification would be propagated to the domain of the remaining variable, so the result of the propagation would be a solution of the original CSP in this case.

Constraint propagation is used to reduce the problem size of CSPs and to reach different levels of local consistency with respect to the value domains of the constraint variables. Another aspect is the application during search algorithms looking for solutions of a specific CSP.

3 Approach

Dependencies between business processes relate the data flow and the input/output interface of processes. We propose the application of AI methods out of the area of knowledge based configuration [Stu97] to build consistent configurations of business processes. A consistent configuration of a business process is twofold: at modelling time it is a consistent static business process model; at runtime it is the consistent state of a business process instance.

We propose how to handle different levels of nested business processes. Flexibility is an important criteria, because different kinds of problem and/or sub-problem dependencies require the flexibility to define different solution strategies and the application of problem specific solving algorithms.

3.1 Ensure Consistent Configurations Using Constraint Satisfaction

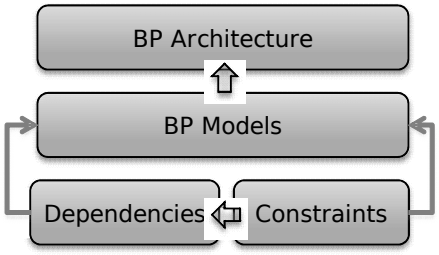

Figure 3: Compliance problem as constraints on dependencies between processes

To ensure consistent configurations of business processes there is the need of management of dependencies between business processes. To create a consistent business process architecture (containing several BP models with dependencies between them), individual requirements for each process have to be satisfied. Usually a process needs some input data to work on and often this input is the result of one or more preceding processes. The results of a process needed by a subsequent one are commonly saved as attribute values. So dependencies between business processes are represented as relations between arbitrary attributes of individual processes. Examples for dependencies are:

1. *sequential dependencies:* Relations between processes in a sequential order. These are relations between the input/output values: the output of a preceding process is needed as input of a subsequent process.

2. *hierarchical dependencies:* One or more processes can be sub-item(s) of a higher-ordered process. Relations between lower and higher-ordered processes. These are relations between the input/output values of the first/last sub-process and the input/output of the higher-ordered process.

An open question is how to handle these dependencies and relations. An answer we propose is a technique out of the field of artificial intelligence (AI), namely constraint satisfaction. Constraint satisfaction is widely used for example in the field of knowledge based configuration, in this case to model relations between components and respectively their attributes [Stu97]. Compliance requirements on processes can be represented as complex relations between the processes. The set of compliance requirements this work deals with are the ones that can be represented as constraints on the dependencies between BP models (see figure 3).

In business process modelling we can use constraints to model relations between the attributes of processes. This means, for algebraic constraints, simple equations and inequations, so called *intensional* constraints.[1] Besides, all types of operators are feasible and fully depend on the implementation of the constraint solver used. Constraints may be used:

- to reduce the possible assignments to variables, which leads to a reduction of problem size,
- for the (early) detection of inconsistencies in the business process model and the business processes at runtime,
- to generate solutions for a certain problem.

Constraint satisfaction in particular supports the propagation of changes to the attribute values and the associated domains throughout a "net of constraints" to reach problem reduction.

The techniques to restrict the domains to a fewer number of values are available for numerous domains (in general finite domains and infinite domains). So the issue to generate consistent configurations of the involved business processes can be handled for combinatorial and numerical problems.

3.2 Static and Dynamic Use of Constraints

The usage of constraint relations for business processes may be a static use at modelling time or dynamic use at runtime.

At modelling time a consistent configuration of a business process is a consistent static process model. The constraints connect input/output variables or attributes of the business processes. Constraint satisfaction is used to check for inconsistencies of the static model like in the following example: Two attributes a and b of two different business processes are connected by the constraint c_1: a < b. The process model would be consistent only if for each value in the value domain of variable a there is a valid value in the domain of b so that the constraint is satisfied. Consistent value domains for example would be the following intervals: a = [0..4], b = [5..9]. Obviously this mechanism can be used for user assistance to optimize problem size of the process model.

At runtime a consistent configuration of a business process is the consistent state of a business process instance. In this case constraint satisfaction is used to check for inconsistencies during the execution of the business processes. User input, external data and calculation results may lead to a reduced solution space.

[1]Which is the implicit definition of a constraint. In contradiction explicit definitions of constraints, e. g. as tables, are called *extensional* constraints.

For example: Given the constraint relation c_2: $a \geq b$ connecting the attributes of two processes. Given further value domains a = [0..9] and b = [0..9], if some input will set b = 5 the value domains can be reduced to a = [5..9] and b = [5]. Further problem reduction may be reached by propagating these changes through the constraint net by constraint propagation. Like in static use this may also show inconsistencies of the current business process configuration.

3.3 Multi-Level Constraint Problem

Different levels of a hierarchy in a business process architecture can be seen as different levels in a hierarchy of constraint problems. Each level of the constraint problem represents a level of the business process architecture. Each business process may consist out of a set of sub-processes, which are sub-problems in the constraint hierarchy.

The resulting hierarchy of processes and constraint relations leads to a *multi-level constraint problem*. The goal using a multi-level constraint problem is to handle different levels of nested business processes. For this task flexibility is needed, because different kinds of problem dependencies require the need to define different solution strategies and the application of problem specific solving algorithms: Different layers in the hierarchy could define different sub-problems. For each sub-problem another solution strategy depending on the value domain of the variables (e. g. finite, infinite, symbolic, Boolean) and the problem structure defined by the constraint net can be applied. At the higher-order levels, the integration of local solutions has to be done in order to achieve globally consistent configurations of business processes.

4 Checking a Business Process Against Constraint Rules

First, we will outline an example for a sequential dependency. A sequential dependency means a constraint has to be satisfied in order that a process is allowed to be the successor of a preceding process. In figure 4 we see *process 1* and *process 2* with the attributes a and b. They are connected via a *constraint connector* which contains the *constraint relation* and the assignments of the attributes to the *constraint variables* in so called *constraint pins*. Process 2 is allowed to be the successor of process 1 only if the constraint relation is satisfied and vice versa: process 1 is allowed to be the predecessor of process 2 only if the constraint relation is satisfied.

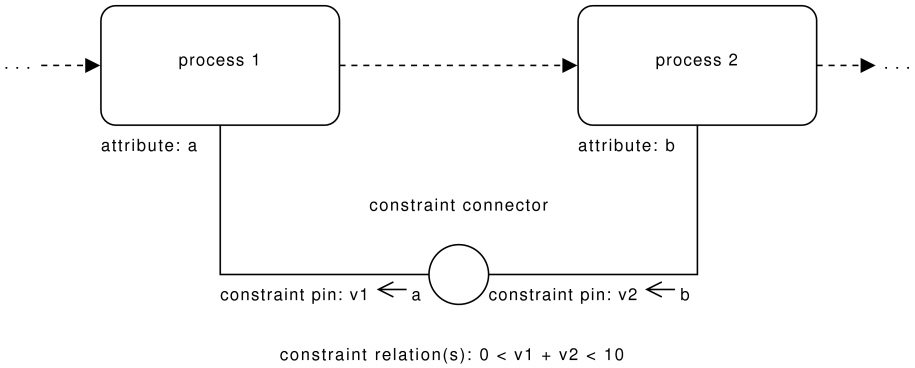

Figure 4: Example for sequential dependency

Our second example is a hierarchical dependency. In figure 5, we have a super-process and two sub-processes and constraint connectors between the processes.[2] The sub-processes are allowed to be sub-items of the super-process only if the constraints hold and vice-versa, the super-process is allowed to be super-item only if the constraints are satisfied. We obtain two levels in a hierarchy of processes and constraint relations, which leads in extension to a multi-level constraint problem.

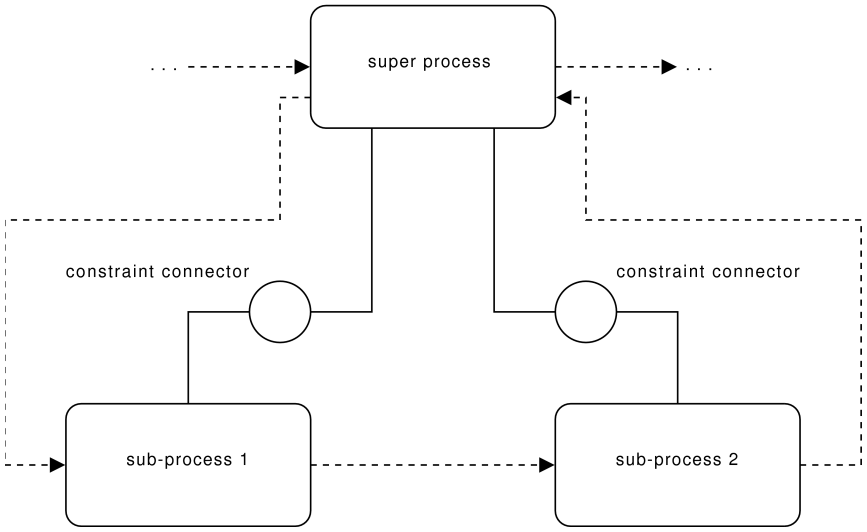

Figure 5: Example for hierarchical dependency

Notice that these techniques can be used at business process design- and runtime.

[2]Details like attributes and constraint relations are left in figure 5 for simplifying reasons.

5 Related Work

The paper proposes the application of constraint satisfaction to ensure the consistency in sequential and hierarchical relations of business processes. Related work to this approach therefore may be found in the domain of constraint satisfaction [Dec03]. Existing constraint satisfaction approaches may be employed to achieve the goals outlined in this paper. Hierarchies of constraint problems can be described by the composite CSP whose main application is in the configuration domain [SF96]. Regarding the aspect of flexibility with respect to problem specific solving algorithms an interesting field is the research on the coordination of cooperative constraint solvers [AM98].

Existing approaches combining business process modelling with constraint satisfaction methods do not focus on different levels in problem hierarchies, solving sub-problems and the integration of solving results. In his approach, Edward P. K. Tsang uses open constraint satisfaction, a branch of constraints research for open-world scenarios, for business process modelling [Tsa03]. Another approach is to use temporal constraint networks to model the temporal relations in business process models [LS+06]. Tools with integrated temporal constraint networks already exist such as workflow management systems [RRD03].

Planning algorithms from the field of AI [RN02] would be an alternative to describe the orderings of sequences of business processes. So another way to handle temporal aspects is the combination of the planning paradigm and the constraint satisfaction paradigm [NN+07]. In this case the planning paradigm is adopted to control the sequence of processes. The constraint satisfaction paradigm is used to keep the objects consistent. It should be considered that the CSP paradigm is sufficient if temporal constraints are used. In our case CSP is employed to handle hierarchies and provide more flexibility in expressing consistency requirements.

Existing application fields of business process modelling in combination with constraint satisfaction methods are the support of workflow collaboration with constraint solving capabilities [CH+04] and dynamic web service composition [CL+05].

6 Conclusion & Outlook

In this position paper we have discussed the problem of compliance in the context of business process management. We have motivated the use of CSP as a formal means for modelling and verifying a certain range of compliance problems. Then the types of problems we tackle are discussed, namely dependencies between business processes. We also provide related thoughts on static and dynamic constraint solving, on multi-level constraint solving as well as on problem solving strategies. Although several work in the context of BPM proposed and provided techniques for using CSP as a tool for verifying constraints on business processes, these approaches mostly focus on the use of one specific technique and do not take a generic approach to the problem. This is the goal of this research. Our aim is to empower compliance and BP experts with a framework for modelling and verifying compliance requirements.

Based on the observations made in Section 2, we plan to extend the fundamental part of this work by conducting a survey of the different kinds of compliance requirements and BP dependencies for which the use of CSP techniques can prove useful. Additionally to this part of the research, a review of the state of the art of CSP techniques will also be conducted. Based on this output, we will study the adequacy of different CSP techniques for the different BP compliance problems. Another important aspect will be a graphical notation for expressing compliance constraints on business processes, which can then be automatically transformed into CSPs. We plan to start validating this work by tackling the modelling and verification of quality of service based selection of services and resources in compositions of business processes.

References

[AHW03] Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske. Business Process Management: A Survey. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske, editors, *Business Process Management, International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003, Proceedings*, pages 1–12, LNCS 2678, 2003. Springer.

[AM98] Farhad Arbab and Eric Monfroy. Coordination of Heterogeneous Distributed Cooperative Constraint Solving. *ACM SIGAPP Applied Computing Review*, 6(2):4–17, 1998.

[CH+04] Yun-Heh Chen-Burger, Kit-ying Hui, Alun D. Preece, Peter M. D. Gray, Austin Tate. Supporting Collaboration Through Semantic-Based Workflow and Constraint Solving. In Enrico Motta, Nigel Shadbolt, Arthur Stutt, Nicholas Gibbins, editors, *Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, EKAW 2004, Whittlebury Hall, UK, October 5-8, 2004, Proceedings*, pages 487–488, LNCS 3257, 2004. Springer.

[CL+05] Nizamuddin Channa, Shanping Li, Abdul Wasim Shaikh, Xiangjun Fu. Constraint Satisfaction in Dynamic Web Service Composition. *Asian Journal of Information Technology*, 4(10): 957–961, 2005.

[Dec03] Rina Dechter. *Constraint Processing*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann Publishers, San Francisco, California, USA, 2003.

[KS+08] Marwane El Kharbili, Sebastian Stein, Ivan Markovic, Elke Pulvermueller. Towards a Framework for Semantic Business Process Compliance Management. In Shazia Sadiq, Marta Indulska, Michael zur Muehlen, editors, *Proceedings of the Workshop on Governance, Risk and Compliance for Information Systems (GRCIS 2008)*, pages 1–15, CEUR Workshop Proceedings Vol-339, Montpellier, France, June 2008.

[Kum92] Vipin Kumar. Algorithms for Constraint Satisfaction Problems: A Survey. *AI Magazine*. 13(1):32–44. 1992.

[LS+06] Ruopeng Lu, Shazia Sadiq, Vineet Padmanabhan, Guido Governatori. Using a Temporal Constraint Network for Business Process Execution. In Gillian Dobbie and James Bailey, editors, *Proceedings of the 17th Australasian Database Conference - Volume 49 (Hobart, Australia, January 16-19, 2006)*, pages 157–166, ACM International Conference Proceeding Series, vol. 170, Darlinghurst, Australia, 2006. Australian Computer Society.

[NN+07] Haruhisa Nozue, Hajime Nakajima, Haruo Oishi, Takeshi Masuda, Tetsuya Yamamura. Process Control Technique Using Planning and Constraint Satisfaction. In Shingo Ata and Choong Seon Hong, editors, *Managing Next Generation Networks and Services, 10th Asia-Pacific Network Operations and Management Symposium, APNOMS 2007, Sapporo, Japan, October 10-12, 2007, Proceedings*, pages 62–71, LNCS 4773, 2007. Springer.

[RN02] Stuart J. Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach (The Intelligent Agent Book)*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2002.

[RRD03] Manfred Reichert, Stefanie Rinderle, Peter Dadam. ADEPT Workflow Management System. In Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske, editors, *Business Process Management, International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003, Proceedings*, pages 370–379, LNCS 2678, 2003. Springer.

[Run09] Wolfgang Runte. Modelling and Solving Configuration Problems on Business Processes Using a Multi-Level Constraint Satisfaction Approach. In Witold Abramowicz, Leszek A. Maciaszek, Ryszard Kowalczyk, Andreas Speck, editors, *Business Process, Services Computing and Intelligent Service Management, Leipzig, Germany, March 23-25, 2009*, pages 237–238, LNI 147, Bonn, Germany, 2009. GI.

[Sch00] August-Wilhelm Scheer. *ARIS - Business Process Modeling*. Springer, 2000.

[SF96] Daniel Sabin and Eugene C. Freuder. Configuration as Composite Constraint Satisfaction. In Boi V. Faltings and Eugene C. Freuder, editors, *Configuration – Papers from the AAAI Fall Symposium*, pages 28–36, Menlo Park, California, USA, 1996. AAAI Press.

[Stu97] Markus Stumptner. An Overview of Knowledge-Based Configuration. *AI Communications (AICOM)*, 10(2):111–125, July 1997.

[Tsa03] Edward P. K. Tsang. Constraint Satisfaction in Business Process Modelling. *The Journal Of Management and Economics*, 7(7), November 2003.