

Constraint-Driven Approach to Support Input Data Decision-Making in Business Process Management Systems

María Teresa Gómez-López, Rafael M. Gasca, Luisa Parody,
and Diana Borrego

1 Introduction

A business process consists of a set of activities that are performed in coordination in an organizational and technical environment (Weske 2007). The base of business process management systems (BPMS) is the explicit representation of business processes with their activities and the execution constraints between them. Compliance rules represent a natural step to include requirements between business functionality and data. For the design of a whole business process management (van der Aalst et al. 2003), it is necessary to design the model of activities and define the causal and temporal relationships between them (Walzer et al. 2008). Compliance rules can help to complete this information, since they can be used to validate business data (Chesani et al. 2008).

Organizations currently need to manage a great deal of data. This can be managed using BPMS, which permit to model tasks and business rules according to a data model. The evaluation of correctness and completeness of values of input data is very important from the point of view of organizations, since neither activity can work correctly using incorrect or corrupted data. But it is especially important when the data are introduced by humans in the instances of a business process.

This chapter takes constraint-driven view of compliance rule engines. We propose to use the constraint programming paradigm to represent and validate the relation between the dataflow variables in compliance rules. It provides a way to find out the possible valid values of the input data that are introduced by humans, in function of the decisions taken in the past, and the compliance rules that have to be satisfiable in the future.

In our proposal, compliance rules are used to analyse the values of the input data and dataflow, in different moments of the business process instance. For this reason,

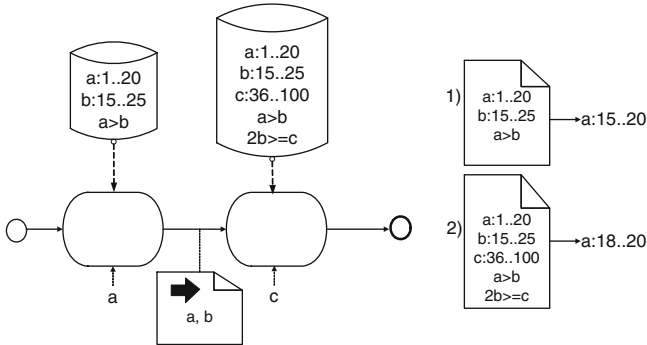


Fig. 37.1 Easy example of business process dataflow for data decision-making support

the use of constraint satisfaction problems for the analysis of the domain of the valid input data is very useful in order to help the user that is introducing input data in the process about the possible correct values of the input variables. The instantiation of the variables, which represent the data in an instantiation process, changes during the process execution; hence, the valid ranges of the variables will change too. In order to introduce the idea of this chapter, in Fig. 37.1 an easy example is depicted, where a very simple business process model is shown. In this process, the value of data a is introduced by humans in the first activity, and the value of variable b is an output value of an activity. In order to decide the valid values of the variable a , we can take into account the compliance rules related only with a and b (option 1), or all the compliance rules related directly or indirectly with a and b (option 2). Option (1) presents the possible values of the variable a if only the compliance rules related to the first activity are analysed, and option (2) shows how the possible valid values of a are reduced if the compliance rules related to the variable c are included in the analysis as well. For example, in option 1, it is possible to introduce the value 16 for the variable a and the value 15 for the variable b , being satisfiable that $\{a > b\}$, but there is no valid values for c to satisfy the compliance rule $\{2b \geq c\}$. Then, the use of compliance rules to represent the relations between dataflow and input variables can help in the introduction of correct values by humans.

The use of compliance rule in a prognosis way can be decisive, since sometimes the humans take decisions in the business processes that result incorrect and when the nonconformities are identified too late. In order to know the valid range of the input variables, we use as compliance rule language and engine the constraint programming paradigm proposed in Teresa Gómez-López and Gasca [2008].

This chapter is organized as follows: Sect. 2 discusses some previous works related to our proposal. Section 3 presents the necessary definitions related to the proposal and an example where decision-making support has been used. Section 4 presents how constraint satisfaction problems can be used in decision-making support. Finally, conclusions and future work are presented.

2 Related Works

The importance of compliance data verification has been the focus of attention for numerous approaches, although most of them have analysed the compliance checking part of process model structure (Sadiq et al. 2005; Ly et al. 2008), or are related to mode checking (Liu et al. 2007; Awad et al. 2008,). Related to how to model data-aware compliance rules, works as Liu et al. [2007], Weber et al. [2008], Ly et al. [2010] and Awad et al. [2011], have defined graphical notations to represent the relationship between data and compliance rules by means data conditions. Also in Knuplesch et al. [2010] a preprocessing step to enable data-aware compliance checking in an efficient manner was presented. Governatori et al. [2008] introduces an approach for semantically annotating activities with preconditions and effects that may refer to data objects, and an efficient algorithm for compliance verification using propagation is also discussed. In contrast to this approach, we focus on analysing the validation on the input data in a business process instance. Most works are related to design and execution business process correctness. Our proposal starts with a correct designed process where an incorrect input instantiation of variables can produce that it works incorrectly.

In zur Muehlen and Indulska [2010], there is a depth analysis about the integration of rules and process modelling and the shortcomings of the existing solutions. Our work is based on Meng [2002], although we propose to separate the evaluation into an independent layer that checks the compliance rules oriented used as a contract that describes the behaviour of the activities in different moments of the business process instance, taking into account the compliance rules evaluated in the future.

The correctness of the values in business processes has been analysed in other proposals, for example, in Rajbhandari et al. [2008], where the “truth degree” of the data can be described by using fuzzy terms. But the mentioned proposal neither allows the monitoring of the process nor the compliance rules analysis. Our proposal is related to the monitoring idea developed in Beeri et al. [2007], which is oriented towards the definition of a monitoring language, although no data decision-making support is presented.

3 Formalization of Data Decision-Making Support for Business Processes

A business process model can be composed by using activities, compliance rules, decision rules and data objects from a process domain. In order to clarify the explanation, we use the well-known example of the organization of a conference, where a reduced model is presented in Fig. 37.2. This business process shows an example where decisions must be taken when there are many unknown variables.

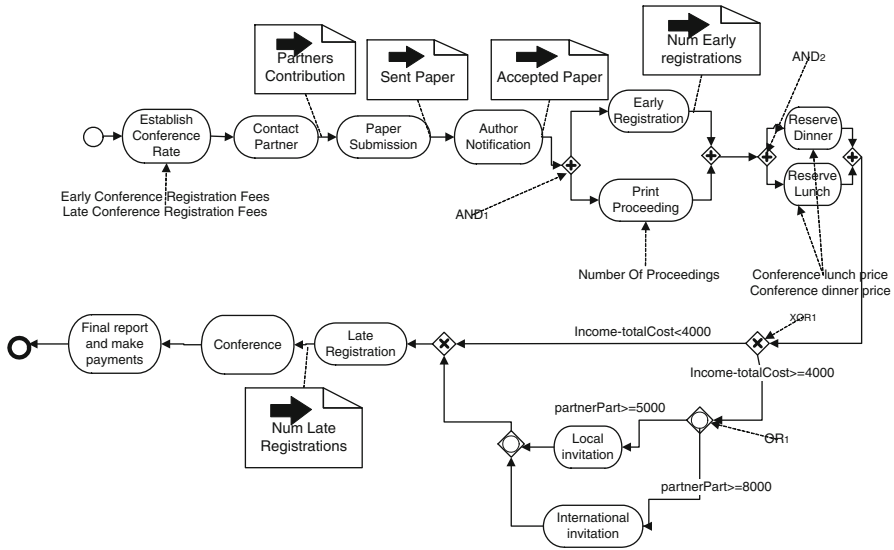


Fig. 37.2 Example of a conference organization process

First of all, the conference committee has to determine the early and late registration fees several months before the number of participants is known, and this decision cannot be changed after the call for paper has been done. Something similar happens to decide the number of proceedings that will be printed, where the final number of participant is unknown. Other decisions, such as to reserve the restaurant for dinner and lunch, have to be taken although the number of participants can have influence in the determination of the price and the restaurant. Although this information is unknown, it is necessary to make a reservation some months before the conference starts. In the example, five different input data participate in the data decision-making process (early conference registration fees, late conference registration fees, conference lunch price, conference dinner price and number of proceeding to print). There are other input variables, but their values are determined in a mandatory way, for example, the number of accepted paper or the final number of participants that belong as part of the dataflow. Obviously, these decisions can be taken for the conference committee, but how the committee can take into account all the possibilities that can happen and the possible branches that will be executed in function of the concrete values of each process instantiation?

The variables related in this example are:

$totalCost$, $Income[0..+\infty]$ is the total cost and income of the conference, respectively
 $numParticipantEarly[50..200]$ is the number of assistant registered in the early period
 $numParticipantLate[10..100]$ is the number of assistant registered in the late period
 $numParticipant[0..+\infty]$ is the final number of participants
 $costPerParticipant[0..+\infty]$ is the cost per participant
 $EarlyRegistrationFees[100..800]$ is price of the early registration
 $LateRegistrationFees[200..1000]$ is price of the late registration
 $incomePartner[1000..10000]$ is the income of the companies whose sponsor the event
 $dinnerPrice[60..250]$ is the price of the dinner
 $lunchPrice[30..100]$ is the price of the lunch
 $proceedingPrice[5..20]$ is the price of print one copy of the proceeding
 $NumOfProceedings[0..+\infty]$ is the number of proceedings printed
 $fixCost[2000..3500]$ is the price of rent the hotel, publicity, cartels. . .
 $inviteSpeaker[0..+\infty]$ is the price of invite a speaker to the conference
 $inviteLocalSpeaker[0..2000]$ is the price of invite a local speaker to the conference
 $inviteInterSpeaker[0..4000]$ is the price of invite an international speaker to the conference

Although the values of the input data are unknown before the conference process starts, the committee has information about previous similar conferences. This information can be represented as constraint compliance rules to help in the datadecision-making process:

$Cost: totalCost = numParticipant * CostPerParticipant + fixCost +$
 $inviteSpeaker + 0.10 * numParticipant * proceedingPrice$
 $numParticipantC: numParticipant = numParticipantEarly$
 $+ numParticipantLate$
 $IncomeC: Income = numParticipantEarly * EarlyRegistrationFeesy$
 $+ numParticipantLate * LateRegistrationFees + incomePartne$
 $RelationPrices:$
 $1.2 * LateRegistrationFees \leq EarlyRegistrationFees$
 $\leq 1.5 * LateRegistrationFees$
 $RelationNumParticipant: 1.2 * numParticipantEarly \geq numParticipant$
 $\leq 1.4 * numParticipantEarly$
 $LunchRelation: costPerParticipant * 0.10 \leq lunchPrice * 3$
 $\leq costPerParticipant * 0.35$
 $DinnerRelation: costPerParticipant * 0.10 \leq dinnerPrice$
 $\leq costPerParticipant * 0.60$

(continued)

(continued)

ProceedingRelation:

$\text{costPerParticipant} * 0.05 \leq \text{proceedingPrice}$
 $\leq \text{costPerParticipant} * 0.15$

CostPerParticipantC: $\text{CostPerParticipant} = 3 * \text{lunchPrice}$
 $+ \text{dinnerPrice} + \text{proceedingPrice}$

InvitationSpeakers: $\text{inviteSpeaker} = \text{inviteLocalSpeaker} + \text{inviteInterSpeaker}$

InvitationLocalSpeakers: $500 \leq \text{inviteLocalSpeaker} \leq 2000$

InvitationInterSpeakers: $1000 \leq \text{inviteInterSpeaker} \leq 4000$

Goal: $\text{Income} * 0.80 \leq \text{TotalCost} \leq \text{Income}$

The constraints *InvitationLocalSpeakers* and *InvitationInterSpeakers* only are used if the activities *Local invitation* and *International invitation* are executed, respectively.

In this chapter, the concrete scenario is defined by means of a process formed by the following:

- \mathbb{A} is a set of activities that defines the model of the process.
- \mathbb{UA} is the set of user activities that is a subset of the activities \mathbb{A} ($\mathbb{UA} \subseteq \mathbb{A}$) that are activities where a user performs the activity with the assistance of a software application and is scheduled, using the terminology proposed in BPMN 2.0 (OMG 2011).
- \mathbb{CF} is a set of control flows (AND, OR, XOR, ...) that describes the relation between the activities.
- \mathbb{DO} is the set of data objects that flows in the process, each of them are defined as an input and/or output data of an activity.
- \mathbb{ID} is the set of input variables introduced by users in the different \mathbb{UA} .
- \mathbb{CR} is a set of compliance rules that can involve dataflow variables and/or input variables of the process used to validate the correctness of the process.
- $\text{Scope}(\mathbb{CR})$ are the activities where \mathbb{CR} can be evaluated after executing them because all the data objects involved in the \mathbb{CR} are instantiated.
- \mathbb{DR} is a set of decision rules associated to the control flows OR or XOR that describes the decisions that the process can take depending of the values of the variables in the dataflow.
- \mathbb{D} is the set of data domains, where $\text{dom}: \mathbb{DO} \rightarrow \mathbb{D}$.

Considering the process example presented in Fig. 37.2, the components of the process following the previous definitions are:

- $\mathbb{A} = \{\text{Establish Conference Rate, Contact Contribution, Paper submission, Author Notification, ...}\}$.
- $\mathbb{UA} = \{\text{Establish Conference Rate, Print Proceeding, Reserve Dinner, Reserve Lunch}\}$.
- $\mathbb{CF} = \text{AND}_1, \text{AND}_2, \text{XOR}_1, \text{OR}_1$.

- $\mathbb{DO} = \{ \text{Output}(\text{Contact Partners}) = \{ \text{Partner Contribution} \}, \text{Output}(\text{Paper submission}) = \{ \text{Sent paper} \}, \text{Output}(\text{Author Notification}) = \{ \text{Accepted papers} \}, \text{Output}(\text{Early Registration}) = \{ \text{num Early Registrations} \}, \text{Output}(\text{Late Registration}) = \{ \text{num Late Registrations} \}.$
- \mathbb{CR} are $\{ \text{totalCost}[0 \dots +\infty], \text{numParticipantEarly}[50 \dots 200], \dots, \text{invitedInterSpeaker}[1000 \dots 4000], \dots, \text{Cost}, \text{numParticipantC}, \dots, \text{Goal} \}$
- Some examples of Scope are: $\text{Scope}(\text{EarlyRegistrationFees}[100 \dots 8000]) = \{ \text{Establish Conference Rate}, \text{Contact Contribution}, \text{Paper submission}, \text{Author Notification}, \dots \}$, $\text{Scope}(\text{InvitationSpeakers}) = \{ \text{Local invitation}, \text{International invitation} \}$ or $\text{Scope}(\text{lunchRelation}) = \{ \text{Reserve lunch} \}$, $\text{Scope}(\text{RelationNumParticipant}) = \{ \text{Late Registration} \}.$
- \mathbb{DR} of the control flow XOR_1 are $\{ \text{Income} - \text{totalCost} < 4000, \text{Income} - \text{totalCost} \geq 4000 \}$, for the control flow OR_1 are $\{ \text{partnerParticipant} \geq 5000, \text{partnerParticipant} \geq 8000 \}.$
- \mathbb{D} for the variables in the example is integer.

3.1 Compliance Rules Representation by Numeric Constraints

Although in the previous section compliance rules have been used, the syntax used in this chapter has not been introduced yet. In order to add compliance rules to a business process related to its data, it is necessary to add semantic information about the data correctness.

If the expressiveness of compliance rules is improved, the mentioned above characteristics will be improved too. For this reason, we propose the use of *Numeric Constraints* instead of *if...then* axiom. The *Constraints* that we propose to define compliance rules can be expressed with the following grammar where *Variable* and *Value* can be defined for integer, natural or float domain:

```

Constraint := Atomic_Constraint BOOL_OP Constraint
            | Atomic_Constraint
            | '¬' Constraint
BOOL_OP := '∨' | '∧'
Atomic_Constraint := function PREDICATE function
function := Variable FUNCTION_SYMBOL function
            | Variable
            | Value
PREDICATE := '=' | '<' | '≤' | '>' | '≥'
FUNCTION_SYMBOL := '+' | '-' | '*'

```

These constraints make easier and more precise to handle numeric data (that can be represented as variables) that represent relations between variables. The use of constraints to represent business rules extends the formal semantics of business

rules, since more knowledge can be inferred and the description has less limitation than using decision trees or a set of facts. For example, it is possible to add the following compliance rule:

$$(a + b + c = d \wedge c \leq a * 0.10) \vee (a + b + c < d \wedge c \leq a * 0.15)$$

where

$$a[1 \dots 100], b[1 \dots 150], c[1 \dots 100], d[5 \dots 250]$$

for Integer domain

By using constraints to represent compliance rules, it is possible to infer new knowledge that is not explicitly described and although not all the related variables (data) are instantiated. Some examples of the inferred compliance rules are:

- $a \leq d, b \leq d, c \leq d$
- $c \leq d * 0.10$
- If $a = 10$ then $d[12 \dots 250] \wedge c = 1$

The knowledge of the third type has great quantity of possibilities, depending of the instantiation of the variables, and it is possible to evaluate a constraint although not all the variables are instantiated. It permits a decision-making task before all the variables are instantiated.

Also, the use of constraints enables integrity rules, derivation rules, reaction rules and production rules to be represented, and the evaluation of whether a set of data is correct for a business policy. The same rule can be of different types depending on the instantiated and known variables. For example:

$$a > b \wedge a + b = c$$

- a, b and c known: It transforms the constraint compliance rules in an integrity business rule.
- a and b known: It transforms the constraint compliance rules in a production business rule.
- a and c known: It transforms the constraint compliance rules in a derivation or reaction business rule, where b is obtained.

It permits to reuse the same compliance rules, avoiding the rewriting of them in different locations of the process and for decision-making support to know the possible valid values of the variables before they are instantiated as it is explained in the next section.

3.2 Dataflow Early Validation Using Compliance Rules

The adoption of business rules adds another tier to systems that automate business processes. Comparing the use of business rules to traditional systems, business rules approach has the following major advantages, analysed in a deeper way in Weber et al. [2009]: lowers the cost incurred in the modification of business logic; shortens development time; rules are externalized and easily shared among multiple applications; changes can be made faster and with less risk.

Although one of the features of compliance rules is the separation of business policy from process control flow, not all the compliance rules have to be related to the whole business process (McDermid 2003), but as it has been commented, it would be a good idea in data decision-making support to analyse the compliance rules that will be involved in the future depending on the model of the process.

We propose to change the initial definition of scope of a compliance rules presented in Sect. 3, checking the valid values before they are instantiated. In this chapter, we change the definition of scope data proposed in Delcambre et al. [2005], where scope data is described as data elements that can be defined which are accessible by a subset of the tasks in a case. We propose to enlarge the scope data definition to compliance rules:

Scope($\mathbb{C}\mathbb{R}$) are the activities where $\mathbb{C}\mathbb{R}$ can be evaluated after execute them because some data objects involved in the $\mathbb{C}\mathbb{R}$ are *related*.

If to know the possible valid values of *Early Registration Fees* and *Late Registration Fees*, only the constraints related to the range of the variables are taken into account, values as *Early Registration Fees* = 120 can be selected. And in future activities, the programme committee will realize that this was a wrong decision. For the example, the domain obtained for *Early Registration Fees* and *Late Registration Fees* are [134 .. 800] and [201 .. 960], respectively, that are obtained analysing not only the compliance rules related to the variables, all the constraints related to the variables related to them.

Then, in each, *User Activity* is necessary to know which compliance rules have to be included in the decision-making process and how the model of the process and the decision rules affect to each decision. In this section, we explain which compliance rules are related to each activity and how the model of the process can affect to the decision problem. How to obtain the valid values is introduced in the next section.

As we have introduced, the scope of a compliance rule is the set of activities where this compliance rule can be evaluated. It means that the compliance rules that can be analysed in each activity (*EvaluatedCR*) are the compliance rules where it is involved, in a formal way as:

If $\text{Scope}(\mathbb{C}\mathbb{R}_1) = \{A_i, \dots, A_j\} \wedge \text{Scope}(\mathbb{C}\mathbb{R}_n) = \{A_k, \dots, A_l\}$
then $\text{EvaluatedCR}(A_i) = \{\mathbb{C}\mathbb{R}_1 \cup \dots \cup \mathbb{C}\mathbb{R}_m\}$
where $A_i \in \text{Scope}(\mathbb{C}\mathbb{R}_1) \vee \dots \vee A_i \in \text{Scope}(\mathbb{C}\mathbb{R}_n)$.

4 Data Decision-Making Support by Using Constraint Programming

As it was commented in Sect. 3, the information between activities is transmitted by dataflow, and this dataflow has to be analysed by compliance rules. Then, it can be used to obtain a reduction of the domains of the variables that, will be valid. In order to know that, we propose the use of constraint satisfaction problems.

Constraint satisfaction problems (CSPs) represent a reasoning framework consisting of variables, domains and constraints. Formally, it is defined as a triple $\langle X, D, C \rangle$ where $X = \{x_1, x_2, \dots, x_n\}$ is a finite set of variables, $D = \{d(x_1), d(x_2), \dots, d(x_n)\}$ is a set of domains of the values of the variables, and $C = \{C_1, C_2, \dots, C_m\}$ is a set of constraints. Each constraint C_i is defined as a relation R on a subset of variables $V = \{x_i, x_j, \dots, x_k\}$, called the *constraint scope*. The relation R may be represented as a subset of the Cartesian product $d(x_i) \times d(x_j) \times \dots \times d(x_k)$. A constraint $C_i = (V_i, R_i)$ specifies the possible values of the variables in V simultaneously in order to satisfy R . Let $V_k = \{x_{k_1}, x_{k_2}, \dots, x_{k_l}\}$ be a subset of X , and an l -tuple $(x_{k_1}, x_{k_2}, \dots, x_{k_l})$ from $d(x_{k_1}), d(x_{k_2}), \dots, d(x_{k_l})$ can therefore be called an *instantiation* of the variables in V_k . An instantiation is a solution only if it satisfies the constraints C .

In order to solve a CSP, a combination of search and consistency techniques is commonly used (Dechter 2003). The consistency techniques remove inconsistent values from the domains of the variables during or before the search. Several local consistency and optimization techniques have been proposed as ways of improving the efficiency of search algorithms.

In this case, the CSPs will be formed by the variables of the dataflow with the restricted domains defined by the expert and/or refined by previous decisions, and the compliance rules represented by numeric constraints that can be evaluated in each moment (EvaluatedCR), but how to create the CSP depends on the topology of the business process model.

As we have commented, all the compliance rules can influence in the possible values of a data, but in which sense will be defined by the topology of the business process model and the compliance rules associated to the control flows. In this chapter, the most common control flows are analysed (AND, OR and XOR). For a sequence of activities or an AND split operator appear, all the constraints related to each of them will be included in the CSP with the AND operator (Fig. 37.3a, b). In the case of XOR control flow, where only one branch can be executed, the constraints of each branch will be combined with the constraint of the control flow and its negation, respectively (Fig. 37.3c). And for OR control flow, each constraint associated to each branch is combined with the constraints of the activities of the branch (Fig. 37.3d).

For the example of Fig. 37.2, the CSP built to analyse the possible valid values is:

```

Income[0..+∞] Integer
totalCost[0..+∞] Integer
...
InviteInterSpeaker[0..4000] Integer
totalCost = numParticipant * CostPerParticipant + fixCost +
inviteSpeaker + 0.10 * numParticipant * proceedingPrice ∧
numParticipant = numParticipantEarly + numParticipantLate ∧
...
Income * 0.80 ≤ TotalCost ≤ Income
(Income - totalCost < 4000) ∨
((Income - totalCost ≥ 4000) ∧
(partnerPart ≥ 5000 ∧ (500 ≤ inviteLocalSpeaker ≤ 2000)) ∧
(partnerPart ≥ 5000) ∧ (1000 ≤ inviteInterSpeaker ≤ 4000))

```

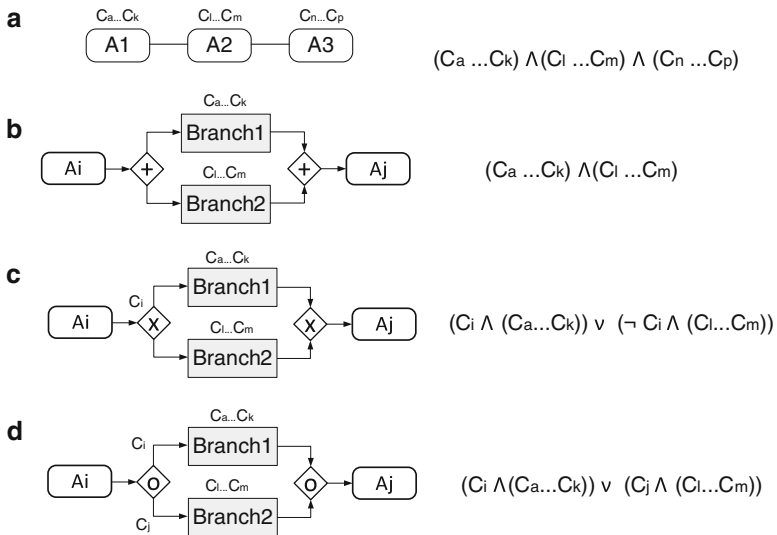


Fig. 37.3 Constraints of the CSPs depending on the business process model

Once the CSP has been solved, the minimum and maximum values for each variable will be obtained. The obtained information can be used to the corresponding business process activity.

5 Conclusions and Future Work

In this chapter the use of constraints to represent compliance rules has been proposed. This representation has been also used to define a decision-making support that provides the valid values of variables when some decisions have to be taken before all the information is known. In order to know the valid values, CSPs have to be created and solved dynamically according to the business model.

These ideas have been developed only taking one instance into account, but the parametrization of compliance rules can be improved learning from previous instantiations. It can also be interesting to provide a way to detect which compliance rules are not satisfiable because their definitions are not correct. Also we propose as future work to extend the type of constraints to handle enumerated domains.

Acknowledgements This work has been partially funded by Junta de Andalucía by means la Consejería de Innovación, Ciencia y Empresas (P08-TIC-04095) and by the Ministry of Science and Technology of Spain (TIN2009-13714) and the European Regional Development Fund (ERDF/FEDER).

References

- Awad A, Decker G, Weske M (2008) Efficient compliance checking using bpmn-q and temporal logic. In: BPM, Milan, pp 326–341
- Awad A, Weidlich M, Weske M (2011) Visually specifying compliance rules and explaining their violations for business processes. *J Vis Lang Comput* 22(1):30–55
- Beeri C, Eyal A, Milo T, Pilberg A (2007) Monitoring business processes with queries. In: VLDB'07: proceedings of the 33rd international conference on very large data bases, VLDB endowment, Vienna, pp 603–614
- Chesani F, Mello P, Montali M, Riguzzi F, Sebastianis M, Storari S (2008) Checking compliance of execution traces to business rules. In: Business process management workshops, Milan, pp 134–145
- Dechter R (2003) Constraint processing. The Morgan Kaufmann series in artificial intelligence. Morgan Kaufmann, San Francisco
- Delcambre LML, Kop C, Mayr HC, Mylopoulos J, Pastor O (eds) (2005) Conceptual modeling – ER 2005, 24th international conference on conceptual modeling, Klagenfurt. Lecture notes in computer science, vol 3716. Springer, Berlin
- Governatori G, Hoffmann J, Sadiq SW, Weber I (2008) Detecting regulatory compliance for business process models through semantic annotations. In: Business process management workshops. Lecture notes in business information processing, vol 17. Springer, Berlin, pp 5–17
- Knuplesch D, Ly LT, Rinderle-Ma S, Pfeifer H, Dadam P (2010) On enabling data-aware compliance checking of business process models. In: ER, Vancouver, pp 332–346
- Liu Y, Müller S, Xu K (2007) A static compliance-checking framework for business process models. *IBM Syst J* 46(2):335–362
- Ly LT, Rinderle S, Dadam P (2008) Integration and verification of semantic constraints in adaptive process management systems. *Data Knowl Eng* 64(1):3–23
- Ly LT, Rinderle-Ma S, Dadam P (2010) Design and verification of instantiable compliance rule graphs in process-aware information systems. In: CAiSE, Hammamet, pp 9–23

- McDermid DC (2003) Integrated business process management: using state-based business rules to communicate between disparate stakeholders. In: *Business Process Management*, Eindhoven, pp 58–71
- Meng J (2002) Achieving dynamic inter-organizational workflow management by integrating business processes, e-services, events, and rules. Ph.D. thesis, Gainesville. Chair-Su, Stanley Y. and Chair-Helal, Abdelsalam
- OMG (2011) *Business process model and notation (bpmn) 2.0 reference manual*
- Rajbhandari S, Rana OF, Wootten I (2008) A fuzzy model for calculating workflow trust using provenance data. In: *Proceedings of the 15th ACM Mardi Gras conference: from lightweight mash-ups to lambda grids: understanding the spectrum of distributed computing requirements, applications, tools, infrastructures, interoperability, and the incremental adoption of key capabilities*, MG '08. ACM, New York, pp 10:1–10:8
- Sadiq SW, Orłowska ME, Sadiq W (2005) Specification and validation of process constraints for flexible workflows. *Inf Syst* 30(5):349–378
- Teresa Gómez-López M, Gasca RM (2008) Run-time monitoring and auditing for business processes data using constraints. In: *International workshop on business process intelligence*, BPI 2010. Springer, Berlin, pp 15–25
- van der Aalst WMP, ter Hofstede AHM, Weske M (2003) *Business process management: a survey*. In: *Business process management*, Eindhoven, pp 1–12
- Walzer K, Breddin T, Groch M (2008) Relative temporal constraints in the rete algorithm for complex event detection. In: *DEBS '08: proceedings of the second international conference on distributed event-based systems*. ACM, New York, pp 147–155
- Weber I, Hoffmann J, Mendling J (2008) Semantic business process validation. In: *SBPM'08: 3rd international workshop on semantic business process management at ESWC'08*, Tenerife
- Weber B, Sadiq SW, Reichert M (2009) Beyond rigidity – dynamic process lifecycle support. *Comput Sci R&D* 23(2):47–65
- Weske M (2007) *Business process management: concepts, languages, architectures*. Springer, New York/Secaucus
- zur Muehlen M, Indulska M (2010) Modeling languages for business processes and business rules: a representational analysis. *Inf Syst* 35(4):379–390