International Journal of Computer and Communication Technology

Volume 6 | Issue 1 Article 7

January 2015

A Structural and Behavioral Analysis Approach for Process Model Evaluation.

SUKANTH SISTLA
Department of CSE JNTUA College of Engineering Ananthapur, India., sukanth.886@gmail.com

A.Ananda Rao
JNTUA COLLEGE OF ENGINEERING ANANTAPUR, ANDHRA PRADESH,INDIA, akepogu@gmail.com

Recommended Citation
SISTLA, SUKANTH and Rao, A.Ananda (2015) "A Structural and Behavioral Analysis Approach for Process Model Evaluation.," *International Journal of Computer and Communication Technology*: Vol. 6 : Iss. 1 , Article 7.
DOI: 10.47893/IJCCT.2015.1269
Available at: https://www.interscience.in/ijcct/vol6/iss1/7

A Structural and Behavioral Analysis Approach for Process Model Evaluation.

[1]SUKANTH SISTLA, [2]PROF.A.ANANDA RAO

Department of CSE
JNTUA College of Engineering
Ananthapur, India.
Email: sukanth.886@gmail.com, akepogu@gmail.com

Abstract-Manufacturing of process driven business applications can be supported by process modeling efforts in order to link the gap between business requirements and system conditions. However, deviating purposes of business process modeling inventiveness have led to considerable problems of aligning related models at distinct abstract levels and distinct outlooks. Verifying the consistency of such related models is a big challenge for process modeling theory and practice. Our contribution is a concept called behavioral profile that sum up the fundamental behavioral limits of a process model. We show that these outlines can be calculated effectively, i.e., in cubic time for sound free-choice Petri nets w.r.t. their number of places and changeovers. In addition to the above Support Vector Machines (SVM) usage is helpful to improve consistency with greater confidence to evaluate behavioral and structural consistency.

Keywords-*change propagation, consistencychecking, behavioral profiles,behavioral equivalence, processmodel*

1 INTRODUCTION

In a Business environment converting the business requirements into a system specification is a difficult task of any software engineering project. To eliminate the gap[2] between business applications and system specifications, business analysts and system analysts have their own perspective needed to be coordinated properly, many applications of such business processing model [1] have given raise to problems and maintaining consistency of such related models has become a challenge for business modeling practice. Behavioral profile is a solution to the inappropriateness of behavioral notions and also change propagation between models including inconsistencies can be resolved. Through this model free-choice Petri nets[4] with reference to their places and transitions, profiles can be computed. Schema integration[5] in particular schema matching investigates and shows such correspondences can be identified automatically. Methodologies for integrated system design like matching techniques and graphical matching can also be applied. Targeting research challenge of defining a notion of consistency between process models[3] is more adequate than existing notions of behavioral equivalence. Behavioral profiles are less sensitive to projections than trace equivalence of as behavioral profiles remain unchanged even if start and end branches are introduced. Profile consistency[1][3] ranging from 0 to 1.0. The proposed change uses Support vector machines can to improve consistency with greater confidence.

Those design decisions may be agreed if they separate from the business process model only to a small degree. Following on the idea of trace equivalence, potential deviations can be counted using a degree of trace consistency calculated based on the ratio of copies of one model that can be mirrored in another model. Still, a relatively small separation in the process model structure (e.g., interchanging two sequential activities) influences on this degree of trace consistency radically. At last, all views of the linear time -branching time spectrum are computationally hard [6]. This is a problem since process models from practice can include easily more than 100activities. This makes the application of trace equivalence and other standard in many interactive modeling situations unrealistic. The official concept of behavioral profile and structural analysis is introduced here. These outlines capture the fundamental behavioral limits of a process model and apply the structural analysis, such as mutual exclusion of activities or partial order. The behavioral profile allows us to overcome three big weaknesses of an application of trace equivalence in an alignment scenario and structural analysis checks the accuracy and consistency measures captured during behavioral outlining.

1) Behavioral outlines are less fragile to projections than trace equivalence. We will show that behavioral outlines of two process models remain unchanged even if additional start and end branches are introduced in one of the models.

2) The structure of a behavioral profile provides us with a straight-forward way to define a degree of consistency ranging from 0 to 1.0, referred to as the degree of profile consistency. In this way, we can feed back detailed information to business analysts and software designers on how far and where two models separate from each other.

3) The theory of a behavioral profile constructs on official properties of free-choice Petri nets. This class of nets has been used for the officialization of most process modeling languages. The source of a behavioral profile and the calculation of a degree of profile consistency and structural analysis of the consistency measured have been implemented to demonstrate the applicability of our approach. In this article, we also report the findings from Verifying consistency between partially overlapping of example process models, a collection of benchmark process models that describe the functionality of specific business software.

2 Associated Work

Here, we discuss four related areas of research, namely Equalities between process models, behavioral equivalence and inheritance, process similarity, and process variability. For the convenience we undertake that Equalities of two process models have been identified and illustrated. As we have seen in the introduction, our view of consistency can be related to desirable properties of schema mappings. Recently, many publications showed how these matching techniques can be applied for business process models [8], [9], [10], [11],[12]. For illustration, [10] uses graph matching techniques in order to identify matching parts of related process models. The thickness of an alignment between process models closely relates to different views of behavioral equivalence, such as trace equivalence and concealment.

Good indications of various equivalence views are presented in [7], [12]. We illustrated the application of the trace equivalence standard in the framework of model projection. With the stimulation by the views of behavioral equivalence, behavior inheritance aspires at applying the idea of inheritance known from static structures to behavioral descriptions. Hare and Kupferman declared that object-oriented system design should merge a notion of behavioral inheritance for classes . The idea to maintain the protocol of a behavioral model is also one of the basic inheritance views by Basten et al. . They define protocol inheritance and projection inheritance based on branded transition systems and division by simulation and mining which aspires at constructing models from event logs.

3 Consistency Measurement Using Behavioral Profiles and Structural Analysis

Business process change is at the very core of business process management, which aspires at enabling flexible adaptation to changing business needs. However, the wide change of drivers for business process modeling inventiveness, reaching from business fruition to process representation, results in number of models that merge in content due to serving different purposes. That, in turn, executes serious challenges for the proliferation of changes between these process models.

Today's Business Process Management (BPM) has a large field of application, reaching from process fruition to process representation. The *purpose* supports the creation of every particular process model. It is a result of this thought that companies create different models for the same process. These models dwell on distinct levels of abstraction and undertake distinct modeling outlooks depending on what is suitable with respect to the modeling goal. The flexibility to adapt business processes in order to react to changing business needs is at the very heart of BPM. Therefore, the *proliferation of changes* between several related process models is a big use case for model alignment. According to Gartner, change is of high significance to the key elements of the BPM discipline, which are *'keeping the business process model in sync with process execution [and] enabling rapid iteration of processes and underlying systems for continuous process improvement and optimization'* .

Proposed system presents a novel approach to change proliferation between business process models. Its central contribution is the definition and application of a technique for dealing with overlapping process models that are not defined in terms of a ordered enhancement. This technique is based on the view of a behavioral profile which sum up a set of dedicated behavioral aspects of a process model. Given a change in the source model, our loom separates a potential *change region* in the target model grounded on the behavioral profile of related activities. In this way, process planned can quickly assess the necessity to propagate the change. If change proliferation seems to be correct, the change region spots the position where to exincline the model.

A. Process Models

Our view of a process model is based on a graph containing activity nodes and control nodes, which, in turn, sum up the commonalities of process description languages.

Thus, the subset of BPMN used in our initial example can be traced back to the following definition of a process model.

B. Behavioral Profile

The Behavioral profile aspires at capturing Behavioral aspects of a process in a finest manner. That is, it consists of three relations between nodes of a process graph. These relations are based on the view of weak order. Two nodes or flow arcs of a process model are in weak order if there a trace in which one node occurs after the other. The existence of only such a trace is required. Thus, weak order does not have to hold for all traces of the model.

- The severe order relation $x \succ P(y)$ *and* $y \neg \succ P(x)$
- The eliteness relation $x \neg \succ P(y)$ *and* $y \neg \succ P(x)$
- The observation concurrency relation $x \succ P(y)$ *and / or* $y \succ P(x)$

The set of all three relations is the Behavioral profile. Two process models with equivalent behavioral outlines may differ in the trace equivalence, in contrast the two process models with identical trace equivalence can also unique in behavioral outlines.

Correspondence Relation: if the relation between two process models is left identical and is not functional

Aligned Changeovers: let a1, a2 correspondence to a and c1, c2 correspondence to c. if transition observed from a1 to c1 ,a1 to c2, a2 to c1 or a2 to c2 then the transition relation between a to c is aligned transition.

Projected Firing sequence: In a arrangement considered, the set of aligned sequences is referred as firing sequence.

Trace Consistency of Alignment: If Aligned exchanges of a projected firing sequence contain trace equivalence then it reproduces as Trace consistency of alignment.

C. Structural analysis:

The structural assessment of dynamically combined process models forms an important step in the model building procedure and it is used for the purpose of the fathomable properties of the model, too. This study contains the determination of the degree of freedom, structural answerability, distinctly index and the dynamic degrees of freedom. As a result of the study, the decomposing of the model is obtained and the calculation path can be found out. This way the suitable numerical method for solving the model can be selected efficiently. Moreover, advice on how to improve the computational properties of the model by modifying its form or its specification can also be given.

Effective graph-theoretical methods have been expected in the literature based on the analysis tools developed by, for the purpose of the most important solvability property of combined dynamic models: the distinction index. The properties of the dynamic representation graph of process models described by semi-explicit DAE-systems have also been analyzed there in case of index 1 and higher index models. Beside the algorithm of determining the distinctly index by using the representation graph, a model modification method has also been assumed in the literature and obtains structurally solvable model even in the case of higher index models.

D. Structural solvability

As a first step, we consider a system of linear or non-linear algebraic equations in its so called *standard form* where $x_j(j=1,...,N)$ and u_k (k = 1,…,K) are unknowns, $y_i(i=1,...,K)$ are known parameters, $f_i(i=1,...,M)$ and $g_k(k=1,...,K)$ are undertaken to be adequately smooth real-valued functions. The system of equations above is structurally solvable, if the Jacobian matrix J(x\ u) referring to the above model is non-singular.

$$y_i = f_i(x,u), i=1,...,M$$
$$u_k = g_k(x,u), k=1,...,K$$

Consider a system of equations in standard form. We construct a directed graph to represent the structure of the set of equations in the following way. The vertex-set related to unknowns and limiting factors is divided as $X \cup U \cup Y$, where $X=\{x_1,...,x_N\}, U=\{u_1,...,u_k\}$ and $Y=\{y_1,...,y_M\}$. The functional dependence described by an equation is expressed by arcs coming into y_i or u_k respectively from those x_j and u_I, which appear on its right-hand side. This graph is called the representation graph of the system of equations.

A Menger-type linking from X to Y is a set of pair- wise vertex-disjoint directed paths from a vertex in X to a vertex in Y. The size of a linking is the number of directed paths from X to Y contained in the linking. In case $|X|=|Y|,(M=N),$ a linking of size X is called a complete linking. The graphical condition of the structural solvability is then the following

Linkage theorem: Undertake that the non-vanishing elements of partial derivatives / and gy, in the standard form model are algebraically independent over the rational number field O. Then the model is structurally solvable if and only if there exists a Menger-type complete linking from X to Y on the representation graph.

We can change the graphical techniques to DAE-systems, as well. An ordinary distinctly equation of a DAE-system can be described by the following equation:

$$x' = f(x_1, ..., x_n)$$

Here ***x*** denotes an random variable depending on time. ***x'*** denotes the derivative of ***x*** with respect to time and $x_1, ..., x_n$ are those variables which have effect oil variable ***x'*** according to the distinction equation.

In DAE-systems there are two types of variables. ***Distinctial variables*** are the variables with their time derivative present in the model. Variables, which do not have their time derivative present, are called ***algebraic variables.*** The derivative ***x'*** is called ***derivative*** (velocity) ***variable.***

E. Dynamic representation graph

The value of distinction variables is actually calculated by using a numerical integration method. Therefore a system of equations including also distinction equations can be represented by a ***dynamic graph.*** A dynamic graph is a series of static graphs related to each time step of the integration. On a dynamic graph there are directed arcs attached from the previous static graph to the next static graph that are find out by the method applied for solving the normal distinction equations. In case of a single step precise method, the value of a distinction variable at time ***t+h*** is calculated using the related distinction value and its value at a previous tune ***t.*** For example, when the explicit Euler method is used:

$$x(t+h) = x(t) + h.x'(t)$$

where ***h*** denotes the step length during the numerical integration. The structure of a dynamic graph supposing exact Euler method for solving distinction equations is shown in ***Fig. 1.***

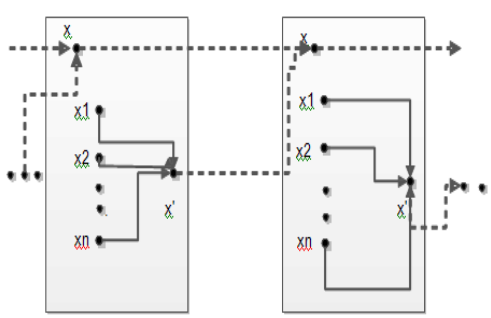


Fig 1: Dynamic Graph model for Euler method for solving distinction

The physical analysis based on graph theoretical technique is carried out in steps sequentially. The first step is to rewrite the model into its standard form. The second step is the obligation of types to vertices in the representation graph. The important types of vertices find outd by the model specification are the following:

- **<S>(set)-type variables*:*** These variables, which are allocated to the specified given values are represented here. In the case of a dynamic representation graph assuming external method for solving the distinction equations, the distinction variables will be labelled by type ***<S*>*** because their starting value can be get from the initial values, and then their values can be calculated step by step by numerical integration. Labels **<*S*>** and **<*S**>** are treated the same way like analysis.
- **<G>(given)-type variables***:* A variable allocated to a specific value of a left hand side is a <G>-type variable. Unlike the <S>-type variables, the values of the right hand side variables will be suitably adjusted so as to preserve the equality of the two sides.

The illustration of graph shows that value of every variable which has incoming arcs only from vertices labeled by type **<*S*>** can be calculated by simple replacement into the related equation. These variables become ***secondarily labeled by type <S>,*** and this process can be looped if necessary. Neglecting all vertices labeled primarily, secondarily, etc. by type **<*S*>** and all arcs starting from them from the illustrating graph we obtain the ***reduced graph.*** The division of vertices of a reduced graph is as follows:

- all initial vertices form the unknown variable set X
- all terminal vertices labelled by type **<*G*>** constitute the known variable (parameter) set Y,
- all other vertices constitute the known variable set ***Y.***

Dynamic process models can be described by ***semi-explicit DAEs*** as follows:

$$z_1 = f(z_1, z_2, t), z_1(t_0) = z_{I0} \quad (1)$$

$$0 = g(z_1, z_2, t) \quad (2)$$

The most important structural computational property of DAE models is the distinction index. By definition the distinction index of the semi-explicit DAE (Equations (l)-(2)) is one if one distinctiation is sufficient to express z_2 as a continuous function of z_1, z_2 and ***t.*** One distinctiation is sufficient if and only if the Jacobian matrix g_{z_2} is non-singular.

In our earlier work we have proved that the distinction index of the models investigated in is equal to 1 if and only if there exists a Menger-type complete linking on the reduced graph at any time step ***t.***

If the distinction index of the investigated model is greater than 1 then there is no Menger-type complete linking on the static graph at any time step ***t.*** The properties of a static graph of a dynamic model, which has distinction index >1 are as follows.

1. The fact that the initial values of distinction variables cannot be chosen individually results in an ***over- specified*** part on the graph. This situation can be easy shown by assignment of types to vertices related to the model specification. There is an over specified part on the graph if a vertex labelled by type **<*S**>** or **<*G*>** can also be labelled preliminary, secondarily, tertiarily or etc. by type **<*S*>.**

Non-singularity of results in an ***underspecified*** part on the graph, hi this part those algebraic variables appear, which cannot be calculated from algebraic equations and those derivative variables, which we want to calculate from them.

We have also predict an algorithm using the structure of the illustration graph for determination of the distinction index of the underlying model. The main steps of tins algorithm are the following:

1. Let us form the following variable sets.

I_0 is the set of the distinction variables belonging to the over specified sub graph,

D_0 is the set of the derivative variables referring to the distinction variables of set I_0,

I_1 is the set of distinction variables from which directed paths lead to the derivative variables in the set D_0,

D_1 is the set of derivative variables referring to the distinction variables of set I_1, ... ,

I_k is the set of distinction variables from which directed paths lead to the derivative variables in the set D_{k-1},

D_k is the set of derivative variables referring to the distinction variables of set I_k ...,

2. Let n be the smallest natural number for which the set ***D„*** contains some derivative variables of the underspecified sub graph. Then the distinction index of the model is

$$v_d = n + 2$$

If there is no such number n then the model is not structurally solvable.

In our previous work we have proved that the important properties of the representation graph including the distinction index of the models are independent of the belief whether a single-step, explicit or implicit numerical method is used for the solution of the distinction equations.

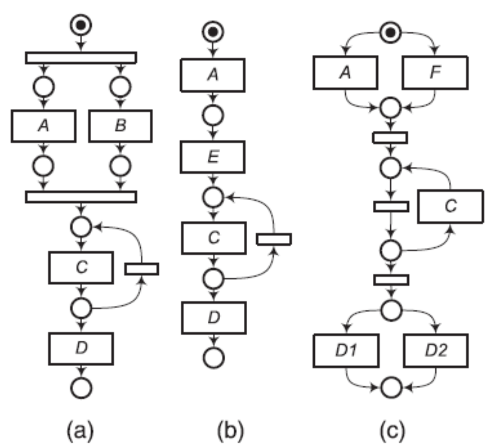
4 CONSISTENCY MEASURES FOR ALIGNED PROCESS MODELS

The previously defined concept of a behavioral profile allows us to officially discuss the view of a degree of profile consistency between a pair of process models. We will use the classical view of trace equivalence, which we exincline to trace consistency, as a benchmark.

4.1 Consistency based on Trace Equivalence

As a benchmark for our consistency analysis, we define a view of consistency based on the trace equivalence standard. First, we adapt the trace equivalence standard for model alignments yielding the view of trace consistency. Second, the degree of trace consistency is introduced based on the amount of traces of one model that have a counterpart in the other model. We already mentioned in Section 2 that the application of trace equivalence in an alignment setting requires that all parts that have been subject to projection are discarded.

4.2 Consistency based on Behavioral Outlines



In general, our view of consistency based on behavioral outlines, i.e., profile consistency, is grounded on the protection of behavioral relations for related activities. In contrast to the view of a trace coherent alignment, it does not require the correspondence relation to be injective. Instead, it allows for 1:n (and even n:m)Equalities. Therefore, this view can be applied to vertical as well as horizontal alignments. Maintenance of the behavioral relation is only required in case there are no merging Equalities. With respect to the examples in Fig. 8 it is easy to see that all pairs of aligned changeovers are also coherent with respect to their behavioral relation. For illustration, the severe order relation between changes. A and D in model 8(a) is preserved for transition pair A and D1, as well as A and D2 in model 8(c). In addition, in all three models it holds C ‖ C. That is, C might occur multiple times during execution.

4.3 Interpretation of Profile Consistency

As demonstrated before, the degree of profile makeup ranges between 0 and 1.0 for two process models and a correspondence relation. Still, a degree of1.0 does not imply that both models are (projected) trace equivalent. This stems from the fact that the underlying behavioral profile represents a behavioral abstraction; in fact, the degree of profile consistency counts the quality of an alignment with respect to the order of potential activity incidence. A degree of 1.0 assure all these limits are equal for the aligned activities of two models. A degree of 0.9, in turn, shows that the limits on the order of potential activity occurrences are equal solely for 90% of the relations between aligned activities. As the degree of profile consistency measures the quality of the alignment, its definition is not dependent of the coverage of the process models by the correspondence relation (i.e., the share of activities in both models that are aligned). Based on the degree of profile consistency, consistency levels might be defined.

However, we undertake these levels to be highly dependent on a specific project setting. Once a degree of profile consistency below 1.0 is observed, the question of how to locate the source of inconsistency has to be addressed. According to our approach, inconsistencies manifest themselves in different relations of the behavioral profile of two process models for a pair of aligned activities. This information can directly be provided to business analysts and system analysts in order to judge on the necessity of the inconsistency. While this kind of feedback allows for locating the inconsistency directly in case of only a few unpredictable profile relations (e.g., caused by an exchanged order of two activities in a sequence), it might be insuitable if a big number of profile relations is unpredictable. Imagine two process models containing a set of aligned activities in sequential order and undertake that one of these activities in one model would now be moved to a branch that is executed together to the remaining activities. Then, all behavioral relations between this activity and the remaining activities would be unpredictable, such that reaction on the set of activities that show relations would be of little help which are unpredictable. Instead, we would consider the biggest subset of aligned activities that show coherent behavioral relations among each other to be valuable feedback on the observed inconsistencies. For the aforesaid case, the single activity having unpredictable relations with all other activities might be identified by this approach.

5 Experiments and Results Analysis:

To analyze the consistency first we have to make the preprocessing of benchmark models. As mentioned before, we setup Equalities between events and functions with equal labels. Further on, we excerpt all pairs of process models that are aligned by at least two Equalities. For such a pair, we then compute the consistency measures, that is, trace consistency, the degree of trace consistency, and the degree of profile consistency of the alignment and finally analyzed the accuracy of the degree of profile using structural analysis.

TABLE 1 :Overall Results

Technique	Precision	Recall	F-Score
Lexical N-M without stemming	0.72	0.60	0.68
Lexical N-M with stemming	0.72	0.60	0.66
A-Star with Post Processing	0.81	0.60	0.69
Greedy	0.89	0.60	0.72

The results are hopeful from the experiments conducted on bench mark business models illustrated in Petri net format. We consider the consistency measurement systems WF systems (WF) [13], and Behavior outlining (BP) analysis[14] to compare with the expected Behavior Outlining and Structural Analysis (BP&SA). We can find the considerable benefit of BP&SA over other models[13,14]. Fig 2 represents the comparison of best in consistency measurement between BP&SA and other two models [13,14]. In fig 3 we can find the computational over head of the WF. Here BP is having slight advantage over BP&SA, which can be negligible while considering the accuracy achieved through BP&SA in consistency measurement.

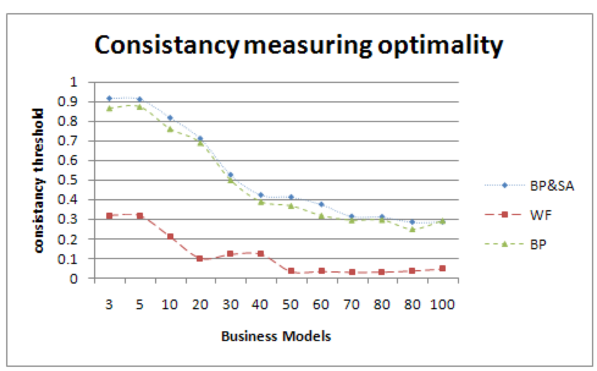


Fig 2: Optimality in Consistency Measurement

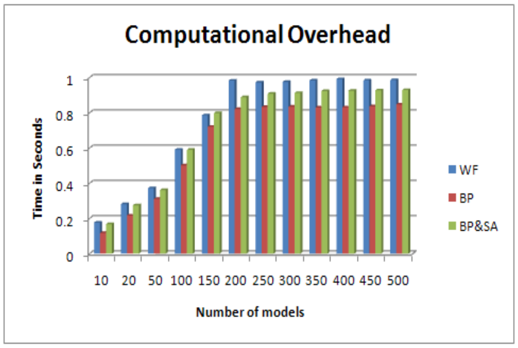


Fig 3: Computational Overhead comparison report

6 CONCLUSIONS

Process models play an vital role to link the gap between business requirements and system conditions. In this paper, we have tested alignment issues between related process models at different abstract levels and different outlooks. More concisely, interlace addressed the research challenge of defining a view of consistency between process models that is more enough to this problem than previous views of behavioral equivalence. The Proposed Concept behavioral profile that sum up the fundamental behavioral limits of a process model. Such behavioral outlines are used for the definition of the official view of profile consistency. Behavioral outlines provide three big rewards opposite to the previous view of trace equivalence and consistency measures that build up it. First, behavioral outlines are less fragile to projections than trace equivalence, as behavioral outlines are fixed even if additional start and end branches are provided. Second, the structure of a behavioral profile provides us with a straight-forward way to define degree of profile consistency between 0 to 1.0 and Structural analysis correct the consistency measurement through degree of profile. At last, the concept of a behavioral profile builds in official properties of free-choice Petri nets. We showed that profile consistency can be checked for sound free-choice WF-systems in O(n3) time with n nodes.

There are number of ways in upcoming research based on behavioral outlines. We have tinted the truth that different interrelated process models and alternatives are used for the growth of process-aware information systems. While we define methods for effectively calculating the behavioral profile, there is currently no easy way back from the profile to a process model. We are hopeful that algorithms can be defined to synthesize process model from a behavioral profile, as there exist synthesis techniques to build Petri nets from transition systems and from traces. Such algorithms might not only take one profile as input. For building merged process models from two behavioral profiles and their alignments currently experiments are going on.

We are currently experimenting with building combined process models from two behavioral outlines and their alignment.

REFERENCES

[1] M. Weidlich, M. Weske and J. Mendling, "Efficient Consistency Measurement Based on Behavioral Profiles of Process Models", IEEE Trans. Software Eng.,vol. 37, no. 3, pp. 410-429, May 2011.

[2] C. Rolland and N. Prakash, "Bridging the Gap between Organisational Needs and ERP Functionality," Requirements Eng., vol. 5, no. 3, pp. 180-193, Oct. 2000.

[3] M. Weidlich, M. Weske and J. Mendling, "Change Propagation in Process Models Using Behavioural Profiles," Proc. IEEE Int'l Conf.Services Computing, 2009.

[4] N. Lohmann, E. Verbeek, and R.M. Dijkman, "Petri Net Transformations for Business Processes—A Survey," Trans. PetriNets and Other Models of Concurrency, vol. 2, pp. 46-63, Springer-Verlag, 2009.

[5] G. Rull, C. Farre and E. Teniente "Validation of Mappings between Schemas," Data and Knowledge Eng., vol. 66,no. 3, pp. 414-437, 2008.

[6] C. Rolland and N. Prakash, "Bridging the Gap BetweenOrganisational Needs and ERP Functionality," RequirementsEngineering, vol. 5, no. 3, pp. 180–193, October 2000.

[7] R. J. van Glabbeek and U. Goltz, "Refinement of actions and equivalence views for concurrent systems," Acta Inf., vol. 37, no.4/5, pp. 229–327, 2001.

[8] S. Nejati, M. Sabetzadeh, M. Chechik, S. M. Easterbrook, andP. Zave, "Matching and merging of statecharts conditions," inICSE. IEEE Computer Society, 2007, pp. 54–64.

[9] R. Dijkman, M. Dumas, L. Garc´ıa-Bauelos, and R. K¨a¨arik, "Aligningbusiness process models," in 13th International IEEE EnterpriseDistributed Object Computing Conference (ECOC), 2009.

[10] M. Weidlich, R. Dijkman, and J. Mendling, "The ICoP framework:Identification of Equalities between process models," inCAiSE, 2010, to appear.

[11] M. Ehrig, A. Koschmider, and A. Oberweis, "Measuring similaritybetween semantic business process models," in ConceptualModelling 2007, Proceedings of the Fourth Asia-Pacific Conference onConceptual Modelling (APCCM 2007), J. Roddick and A. Hinze, Eds.,vol. 67. Ballarat, Victoria, Australia: Australian Computer ScienceCommunications, 2007, pp. 71–80.

[12] Dling, "MeasuringSimilarity between Business Process Models," in CAiSE, ser.Lecture Notes in Computer Science, Z. Bellahsene and M. L´eonard,Eds., vol. 5074. Springer, 2008, pp. 450–464.

[13] W. van der Aalst, M. Dumas, F. Gottschalk, A. ter Hofstede, M. L.Rosa, and J. Mendling, "Correctness-preserving configuration of business process models," in Proc. of Fundamental Approaches toSoftware Engineering (FASE 2008), ser. Lecture Notes in ComputerScience, J. L. Fiadeiro and P. Inverardi, Eds., vol. 4961. Springer,2008.

[14] Weidlich, M.; Mendling, J.; Weske, M.; , "Efficient Consistency Measurement Based on Behavioral Outlines of Process Models," Software Engineering, IEEE Transactions on , vol.37, no.3, pp.410-429, May-June 2011

◈◈◈