For Industry Day

A Conformance Checker tool CSPConCheck

Suman Roy[1], Sidharth Bihary[1], and Jose Alfonso Corso Laos[2]*

[1] Infosys Lab, Infosys Ltd.,
44 Electronics City, Hosur Road, Bangalore 560 100
Email:{Suman_Roy, Sidharth_Bihary}@infosys.com
[2] College of Engineering, Pontificia Universidad Catolica De Chile,
Libertador Bernardo OHiggins, 340-Santiago-Chile
Email: pepecorso@gmail.com

Abstract. We describe a tool on conformance checking which verifies if the event logs (observed) match/fit the reference (arbitrary) process, we call this tool "CSPConCheck". We use concepts from Communicating Sequential Processes (CSP), which facilitates automated analysis using PAT toolkit for conformance checking. This tool takes input process diagram and process logs and uses PAT tool to check for conformance. Further, this tool is capable of computing some metrics related conformance checking.

1 Introduction

Process-aware information systems are widely used in industries these days as they provide precise description of business requirements; industrial regulatory business activities need to be monitored for auditing an organization (which is made mandatory by the new Sarbanes-Oxlay (SOX) Act [6]) in conjunction with business process modeling and simulation. A natural question arises - how closely the observed behaviour follow or fit the specified behaviour. This is known as the problem of "*conformance*". This tool aims at conformance checking of business processes which can result in balancing 'Business-IT' platform in software development thereby improving the processes of correct simulation, requirement elicitation etc.

Information systems, such as WFM, ERP, CRM, SCM and B2B systems maintain foot prints of execution of the activities, which is also called event logs or transaction logs. Each event log denotes one instance of the process. In this work, we follow Business Process Modeling Notation (BPMN) to capture the processes, which are modeled using an in-house tool, called InFlux. BPMN allows four types of constructs: event, activity, control-flow gateway and sequence. An event denotes the position of the execution of the process. Two types of events are considered, *start event* and *end event* which represents the initiation and termination of the process respectively. The event logs of the model are recorded in MXML files which provides the execution sequence of the activities, number of instances of the log and some other data.

* This work was done when the author was an intern with Infosys Lab during Dec'11-Feb'12.

We use a process algebraic language called *Communicating Sequential Process* (*CSP*) for generating compact representations of both the reference model and the event logs. As said earlier, CSP is a process algebra for describing processes or programs [3, 5] in which processes communicate via events. The CSP for the reference model is called the specification *(Spec)*. We consider the event logs and generate CSP, which is called implementation *(Impl)*. Both spec and impl are trace equivalent with the process and event logs respectively. Then CSP processes are fed into PAT model checker [7, 4] and using the notion of trace refinement, it is checked whether the *Impl* trace refines *Spec*.

Actually, we implemented the above described algorithms with PAT tool [7] at the back end. We call this tool "CSPConCheck". The algorithms for generating CSP descriptions for the processes is a modification of the algorithm used for converting UML diagrams to CSP descriptions [2]. The basic methodology of the mapping is to decompose the process model into atomic patterns and generate independent CSP description for each of the patterns. Then using the prefix process the CSPs are merged to get the complete CSP description of the whole process. For activity and event nodes the CSP is a simple prefix process. For gateway nodes it uses different operators depending upon the type of the gateway. The nodes of the process model are considered as events in the CSP description and the edges of the model are treated as processes. The CSP description for an event log is generated by constructing a single CSP process for each trace in the log set and then aggregating all of them using external choice operator. The CSP for each trace is a simple prefix process as the traces contain only the sequence of activities [1]. When both the CSP descriptions are fed to the tool it decides whether the logs conform to the process or not. In case of the latter, by using the shortest counterexample produced by PAT tool our tool can produce all the error traces (an error trace is one which cannot replayed on the process). Moreover, we added a graphical interface (the Editor module) which shows the user all the erroneous traces. Also this tool can compute metrics related to conformance checking that shows the fitness, closeness, and appropriateness of the event logs with respect to the reference process models.

2 Description of CSPConCheck

CSPConCheck is a JAVA based tool. The tool allows the user for GUI based selection option of the folder where the reference model (in XMI format) and the event log (in MXML format) are saved. We use our in-house process modeler (InFlux tool) to model the process which generates an XMI file. The event logs are assumed to be in MXML format. The tool picks up the process models and relevant logs for further processing. When the application is run, it generates one CSP file against each model which contains the CSP descriptions for the reference model and for the corresponding event logs in machine readable CSP. We have written Java code to convert XMI representation of processes to CSP descriptions in machine readable format of PAT tool. Similarly, another Java code is used to translate event logs to machine readable CSP description. Both these CSP descriptions are fed into PAT toolkit which is a windows based tool. The tool shows the result for conformance checking. A schematic diagram of this

setup is shown in Figure 1. The tool offers a GUI for the visualization of the reference model and replay of logs in an interactive manner. CSPConCheck has a Metric module which computes and displays the metric values for reference model-event log set which shows how closely the event log is aligned to the model. In case of erroneous traces, the tool has a Display module which shows all the non-complied traces.

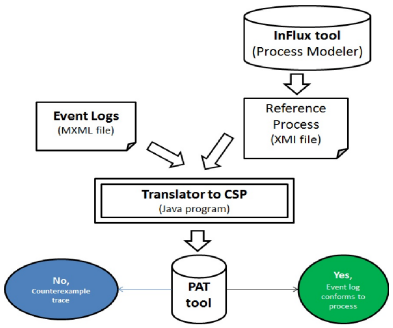

Fig. 1. A schematic diagram of the tool

3 Functionalities of CSPConCheck

In this section we will illustrate the funtionalities of the tool though a case study. Figure 2 is a typical example of a business process followed in a bank for opening account, modeled using Influx tool. The process is free of control flow related errors like deadlock and lack of synchronization, and contains complex constructs like loops, unstructured gateways etc. We use the convention that an activity is labeled with an alphabet displayed against it. Table 1 shows the event

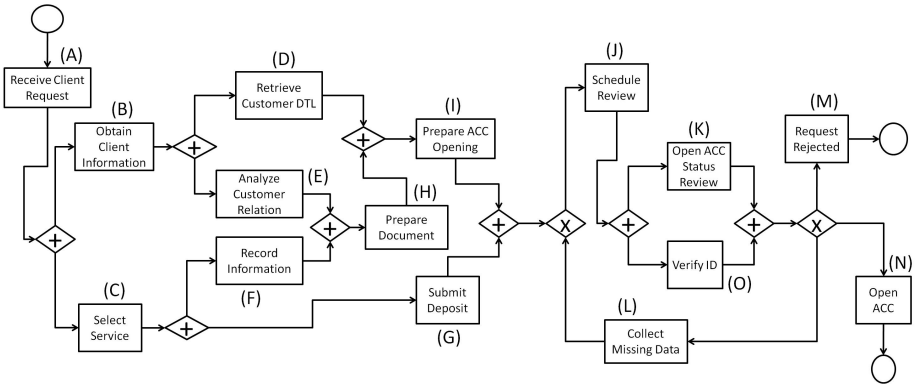

Fig. 2. An example of a bank account opening process

logs for the process which comprise of sequence of activities i.e. traces and the corresponding instances of occurrence.

No of instances	Log Traces
92	ACBEGFDHIJKOL
34	ACBGFEDHIJKOLJKOM
81	ABCDEGFHIJLKOJKOM
9	ABECHFGDIJKON
80	ACBDEGFHIJKOLJKON
2	ABCEGDFHIJKOLJKOLJKON

Table 1. Event logs for bank account opening process

When the CSPs for the reference model and the event logs are generated using CSPConCheck and are fed into PAT toolkit, the tool shows that the event log does not conform to the reference model and reports the shortest counter example as $< A, B, E, C, H >$. Our tool can find all the traces with the counter example as prefix and display the complete set of erroneous traces which cannot played on the reference model. Thus CSPConCheck displays $\{ACBEGFDHIJKOL, ABCDEGFHIJLKOJKOM, ABECHFGDIJKON\}$. Subsequently it computes all the metrics for conformance checking and displays in the Metric module. Fitness metric measures how closely the event log is aligned to the reference model i.e. even in the case of erroneous traces how far the trace can be replayed in the model. Similarly the closeness metric denotes what fraction of the instances actually conform to the model. For a log that fully conforms to the model, both the metric values must be 1. For the case study example the values for fitness metric and closeness metric come out to be 0.576 and 0.389 respectively. Further the tool calculates appropriateness metrics (both structural and behavioral). Structural appropriateness deals with the size of the model and denotes how compactly the process has been designed where as behavioral appropriateness captures behavioral difference between the event logs and the reference model. Though an event log set conforms to a reference model, the model can show more behavior resulting in being less informative about the event logs. For the example, the value for structural appropriateness metric is 0.68 while the behavioral appropriateness comes out to be 0.042, which means the behavior of the event logs portrays very little account of the behavior of the reference model.

References

1. S. Bihary, J. Koneti, and S. Roy. Process conformance using csp. In *Proceeding of the 5th ISEC'12*, pages 139–142. ACM, 2012.
2. D. Bisztray and R. Heckel. Rule-level verification of business process transformations using CSP. *ECEASST*, 6, 2007.
3. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
4. Y. Liu, J. Sun, and J. S. Dong. Pat 3: An extensible architecture for building multi-domain model checkers. In *IEEE 22nd International Symposium on Software Reliability Engineering, (ISSRE'11)*, pages 190–199, 2011.
5. A. W. Roscoe. *The theory and practice of concurrency*. Prentice Hall, 1997.
6. P. Sarbanes, G. Oxley, and et. al. Sarbanes-oxley act of 2002, 2002.
7. J. Sun, Y. Liu, A. Roychoudhury, S. Liu, and J. S. Dong. Fair model checking with process counter abstraction. In *Proceedings of the 2nd World Congress on Formal Methods*, FM '09, pages 123–139. Springer-Verlag, 2009.